

**Customer Engineering  
Manual of Instruction**

**1410**

**Data Processing System**

# CONTENTS

1. 0. 00	SYSTEM FUNDAMENTALS . . . . .	5	2. 7. 00	Indexing . . . . .	77
1. 1. 00	Storage Principles . . . . .	5	2. 7. 01	Index Register Locations . . . . .	77
1. 1. 01	Addressing . . . . .	5	2. 7. 02	Index Tags . . . . .	78
1. 1. 02	Data Word . . . . .	5	2. 7. 03	Operation . . . . .	78
1. 1. 03	Variable Word Length . . . . .	6	2. 8. 00	Data Handling Op Codes . . . . .	80
1. 1. 04	Word Mark . . . . .	6	2. 8. 01	Op Code Function . . . . .	80
1. 1. 05	Scanning . . . . .	6	2. 8. 02	Move Character and Suppress Zero Op Code . . . . .	83
1. 2. 00	Stored Programming . . . . .	6	2. 8. 03	Compare Op Code . . . . .	87
1. 2. 01	Instruction Word . . . . .	6	2. 8. 04	Edit . . . . .	88
1. 3. 00	Control . . . . .	7	2. 9. 00	Arithmetic Op Codes . . . . .	96
1. 3. 01	Instruction Phase . . . . .	8	2. 9. 01	Add or Subtract . . . . .	96
1. 3. 02	Execute Phase . . . . .	9	2. 9. 02	Zero and Add or Zero and Subtract . . . . .	100
1. 4. 00	Data Flow . . . . .	9	2. 9. 03	Multiply . . . . .	101
1. 4. 01	Control Processing Unit . . . . .	9	2. 9. 04	Divide . . . . .	112
2. 0. 00	CENTRAL PROCESSING UNIT . . . . .	13	2. 10. 00	Logic Op Codes . . . . .	118
2. 1. 00	Core Storage . . . . .	13	2. 10. 01	Branch Operation . . . . .	118
2. 1. 01	Ferrite Cores . . . . .	13	2. 10. 02	Branch if Internal Indicator On . . . . .	118
2. 1. 02	Core Plane . . . . .	13	2. 10. 03	Branch if I/O Channel-Status Indicator On . . . . .	120
2. 1. 03	Core Array . . . . .	14	2. 10. 04	Branch if Character Equal . . . . .	122
2. 1. 04	Data Flow Components . . . . .	18	2. 10. 05	Branch if Bit Equal . . . . .	124
2. 1. 05	Matrix Switching . . . . .	21	2. 10. 06	Branch on Word Mark or Zone Equal . . . . .	124
2. 1. 06	Noise Problems . . . . .	24	2. 11. 00	Miscellaneous Op Codes . . . . .	125
2. 1. 07	1411 Core-Storage Unit . . . . .	26	2. 11. 01	Table Look-Up . . . . .	125
2. 1. 08	Core-Storage Addressing . . . . .	26	2. 11. 02	Store Address Register . . . . .	127
2. 1. 09	Memory Clock . . . . .	29	2. 11. 03	Set Word Mark . . . . .	130
2. 1. 10	Character Gating . . . . .	32	2. 11. 04	Clear Word Mark . . . . .	132
2. 2. 00	Address Unit . . . . .	33	2. 11. 05	Clear Storage . . . . .	135
2. 2. 01	Address Registers . . . . .	33	2. 11. 06	Clear Storage and Branch . . . . .	136
2. 2. 02	Address Modification . . . . .	35	2. 11. 07	Halt . . . . .	136
2. 2. 03	Address Exit Channel-Validity Check . . . . .	38		Halt and Branch . . . . .	137
2. 2. 04	Address-Channel Validity Check . . . . .	38		No Operation . . . . .	137
2. 3. 00	Control Unit. . . . .	40	3. 0. 00	INTEGRATED SYNCHRONIZER . . . . .	138
2. 3. 01	Cycle Control . . . . .	40	3. 1. 00	Card Punch . . . . .	138
2. 3. 02	Logic Check . . . . .	41	3. 1. 01	Clock . . . . .	140
2. 4. 00	Operational Unit . . . . .	44	3. 1. 02	Core Storage . . . . .	140
2. 4. 01	A-Data Register . . . . .	44	3. 1. 03	Punch-Storage Scanning . . . . .	142
2. 4. 02	B-Channel Validity Check . . . . .	44	3. 1. 04	Punch Decode and Punch Matrix . . . . .	144
2. 4. 03	A-Channel . . . . .	47	3. 1. 05	Punch-Move or Load Op Code . . . . .	147
2. 4. 04	Assembly . . . . .	47	3. 2. 00	Card Reader . . . . .	161
2. 4. 05	Adder . . . . .	52	3. 2. 01	Core Storage . . . . .	163
2. 4. 06	Compare . . . . .	53	3. 2. 02	Read-Storage Scanning and Priority Controls . . . . .	163
2. 5. 00	Resets . . . . .	57	3. 2. 03	Read Encode . . . . .	167
2. 6. 00	Instruction Read-Out . . . . .	63	3. 2. 04	Stacker Select and Feed Op Code . . . . .	167
2. 6. 01	Operation . . . . .	64	3. 2. 05	Reader Move or Load . . . . .	174
2. 6. 02	Chaining . . . . .	69	3. 3. 00	Paper-Tape-Reader Attachment . . . . .	193
2. 6. 03	Op Register Set Check . . . . .	70	3. 3. 01	Core Storage . . . . .	193
2. 6. 04	Op-Modifier Register Check . . . . .	70			
2. 6. 05	Instruction Check . . . . .	70			
2. 6. 06	Address Check . . . . .	75			

3. 3. 02	Input-Serial Scan Control . . .	195	7. 1. 13	Console Display . . .	318
3. 3. 03	Paper-Tape Read-In . . .	195	7. 1. 14	Alter Routine . . .	322
3. 3. 04	Paper-Tape Reader Move or Load . . .	197	7. 1. 15	I/E Mode . . .	326
3. 4. 00	Checking Features . . .	213	7. 1. 16	CE Controls . . .	326
3. 4. 01	Parity Check . . .	213			
3. 4. 02	Ring Check . . .	213	8. 0. 00	IBM 1401 COMPATIBILITY . . .	333
3. 4. 03	Clock Check . . .	213	8. 1. 00	Zone and Auxiliary Binary Adders . . .	333
3. 4. 04	Hole-Count Check . . .	215	8. 2. 00	Instruction Read-Out . . .	335
3. 4. 05	Reader Validity Check . . .	223	8. 3. 00	Indexing . . .	340
			8. 4. 00	Data Handling . . .	344
4. 0. 00	PRINT STORAGE . . .	225	8. 4. 01	Move Numerical . . .	345
4. 1. 00	Printer Control . . .	225	8. 4. 02	Move Zone . . .	345
4. 1. 01	Printer Core-Storage Unit . . .	225	8. 4. 03	Move Characters to Record or Group Mark . . .	347
4. 1. 02	Type Synchronization . . .	231	8. 4. 04	Move Characters to A or B Word Mark . . .	349
4. 1. 03	Print-Equal-Compare and Hammer Fire . . .	236	8. 4. 05	Load Characters to A Word Mark . . .	349
4. 1. 04	Printer Move-Op Code . . .	236	8. 4. 06	Move Characters and Suppress Zeros . . .	350
4. 2. 00	Carriage Control . . .	246	8. 4. 07	Compare . . .	350
4. 2. 01	Carriage Circuits . . .	246	8. 4. 08	Move Characters and Edit . . .	350
4. 2. 02	Forms Control Op Code . . .	248	8. 5. 00	Arithmetic . . .	350
4. 3. 00	Checking Features . . .	254	8. 5. 01	Add and Subtract . . .	350
4. 3. 01	Printer Synchronization Check . . .	254	8. 5. 02	Zero and Add, Zero and Subtract . . .	351
4. 3. 02	Printer Parity Check . . .	254	8. 5. 03	Multiply and Divide . . .	351
4. 3. 03	Print-Error Detector . . .	257	8. 6. 00	Logic Operations . . .	351
			8. 6. 01	Branch . . .	351
5. 0. 00	MAGNETIC TAPE CONTROL . . .	259	8. 6. 02	Branch, if Indicator On . . .	352
5. 1. 00	Magnetic-Tape-Operation Instructions . . .	259	8. 6. 03	Branch, if Character is Equal . . .	352
5. 1. 01	Tape Write Operation . . .	259	8. 6. 04	Branch, if Bit Equal . . .	357
5. 1. 02	Tape Read Operation . . .	263	8. 6. 05	Branch, if WM and/or Zone . . .	357
5. 1. 03	Tape-Unit Control Operation . . .	266	8. 7. 00	Miscellaneous . . .	358
			8. 7. 01	Store A-Address Register . . .	358
6. 0. 00	DISK-STORAGE CONTROL . . .	269	8. 7. 02	Store B-Address Register . . .	358
6. 1. 00	Disk-Storage Control Instructions . . .	269	8. 7. 03	Modify Address . . .	362
6. 1. 01	Seek-Disk Record . . .	269	8. 7. 04	No Operation . . .	368
6. 1. 02	Disk-Address Transfer . . .	273	8. 7. 05	Halt . . .	368
6. 1. 03	Disk Write . . .	279	8. 7. 06	Halt and Branch Op . . .	370
6. 1. 04	Disk Read . . .	282	8. 7. 07	Set Word Mark . . .	370
6. 1. 05	Disk-Write Address . . .	286	8. 7. 08	Clear Word Mark . . .	370
			8. 7. 09	Clear Storage . . .	371
7. 0. 00	IBM 1415 CONSOLE AND INPUT-OUTPUT PRINTER . . .	287	8. 7. 10	Clear Storage and Branch . . .	371
7. 1. 00	Console Functional Units . . .	293	8. 8. 00	Input-Output Instructions . . .	373
7. 1. 01	Console Clock . . .	293	8. 8. 01	Read a Card . . .	373
7. 1. 02	Console Cycle-Control Matrix . . .	293	8. 8. 02	Punch a Card . . .	382
7. 1. 03	Console Cycle-Control Matrix Advance . . .	293	8. 8. 03	Print a Line . . .	389
7. 1. 04	Character-Print Operation . . .	294	8. 8. 04	Write Word Marks . . .	396
7. 1. 05	Space Operation . . .	295	8. 8. 05	Combination Card-Print Instructions . . .	398
7. 1. 06	Output Error Routine . . .	295	8. 8. 06	Select Stacker . . .	398
7. 1. 07	Word-Mark Control . . .	299	8. 8. 07	Carriage Control . . .	402
7. 1. 08	Carriage Return . . .	300	8. 8. 08	Card-Print-Branch Instruction . . .	406
7. 1. 09	Stop Print-Out . . .	300	8. 8. 09	Stacker Select or Control . . .	407
7. 1. 10	Inquiry . . .	304			
	Read Console Printer . . .	305			
7. 1. 11	Write Console Printer . . .	310			
7. 1. 12	Console Address Set . . .	315			

8.8.10	Start Read Feed and Start Punch Feed . . . . .	409	9.2.11	Printer-Selected Off-Line Mode . . .	435
8.8.11	I/O Error Conditions . . . . .	409	9.2.12	Oscilloscope Sync Generation . . .	437
			9.3.00	Tape-Adapter Unit CE Panel . . .	437
9.0.00	SYSTEM CE CONTROLS . . . . .	413	10.0.00	POWER SUPPLIES . . . . .	443
9.1.00	IBM 1411 CE Console . . . . .	413	10.1.00	Regulation . . . . .	443
9.1.01	Address Display . . . . .	413	10.2.00	Special-Voltage Power-Supply Unit . .	443
9.1.02	B-Character-Select Switch . . . . .	413	10.3.00	Power-On Sequence . . . . .	446
9.1.03	Portable Controls . . . . .	413	10.4.00	Emergency Off . . . . .	446
9.1.04	Address Stop and Scope Sync . . . .	413	10.5.00	DC-Off . . . . .	446
9.2.00	IBM 1414 I/O Synchronizer CE Panel .	415	10.6.00	Power-Off . . . . .	450
9.2.01	Switches, Keys, and Lights . . . . .	415	10.7.00	Local Remote Operation -10 . . . .	450
9.2.02	Read-to Punch Off-Line Mode . . . .	419	10.8.00	TAU Off-Line Operation-Local-Remote Switch . . . . .	451
9.2.03	Read-to-Print Off-Line Mode. . . . .	421	10.9.00	File Control Unit (FCU) Off-Line Operation Local-Remote . . . . .	451
9.2.04	Card-Read Integrated-Buffer Off-Line Mode . . . . .	423	10.10.00	Power Fault Indicators . . . . .	451
9.2.05	Card Punch Integrated-Buffer Off-Line Mode . . . . .	425	10.10.01	Thermal Reset . . . . .	451
9.2.06	Paper Tape Read-Integrated-Buffer Off-Line Mode . . . . .	427	10.10.02	Overcurrent Failure . . . . .	451
9.2.07	Print Integrated-Buffer Off-Line Mode .	429	10.10.03	Overvoltage Failure . . . . .	451
9.2.08	Card Reader Selected Off-Line Mode .	431	10.11.00	Marginal Checking . . . . .	451
9.2.09	Card-Punch-Selected Off-Line Mode .	433	10.12.00	Portable Marginal-Check Power-Supply Unit . . . . .	453
9.2.10	Paper-Tape-Reader-Selected Off-Line Mode . . . . .	433	10.13.00	Power-Unit Locations . . . . .	455
			10.13.01	Frame and Chassis Designations . .	455



## 1.0.00 SYSTEM FUNDAMENTALS

### 1.1.00 STORAGE PRINCIPLES

Magnetic core storage is used in the IBM 1411 Central Processing Unit. Characters are stored in the core-storage unit in binary-coded decimal (BCD) form. This requires seven bit locations, or core planes (C, B, A, 8, 4, 2, 1) for each character position. An eighth core plane provides for the storage of word marks (see the section 1.1.04 Word Mark).

#### 1.1.01 Addressing

Core-storage units are available in 10,000-, 20,000-, or 40,000-character position capacities. The arrays are arranged so that each character position can be individually addressed.

Each core-storage address is five character positions. Valid addresses for a machine with 10,000 characters of storage are 00000 to 09999. Valid addresses for a machine with 20,000 characters of storage are 00000 to 19999. Valid addresses for the maximum machine with 40,000 characters of storage are 00000 to 39999.

An attempt to use an invalid address results in an error. The numerical portion of the five-character address must consist only of valid numerical numbers from zero to nine. An attempt to use 8-3, 8-4, 8-5, 8-6, 8-7 or blank codes in addresses results in an error-stop (address validity). The zone bits over the tens and hundreds positions are reserved for index tags as explained under 2.7.00 Indexing. The units-, thousands-, and ten-thousands-positions must contain a no-zone indication, or an error-stop results.

Core-storage addressing is as follows:

1. To address an instruction in core storage, specify the location of the high-order character (operation code).
2. Fields to be transferred within the core storage, or fields to be operated on arithmetically, are addressed by specifying the location of the low-order character of the field or record. Characters are transferred or operated on arithmetically in a low- to high-order sequence.
3. For all data movements that originate or terminate outside the core storage, and for record moves within core storage, the data is addressed by specifying the location of the high-order character. Therefore, data addresses of all input-output operations, including the file, specify the location of the high-order character. On an I/O operation, characters are moved from high to low order.

#### 1.1.02 Data Word

A data word is one complete unit of information that is comparable to a field in a card, such as an account number.

### 1.1.03 Variable Word Length

A data word may be a single character, or a group of characters. Words are not limited to any fixed number of character positions in the storage unit.

### 1.1.04 Word Mark

To define the length of a word, a word mark is stored as a single bit in the eighth core plane of storage in the high-order position of that word. This word-mark core-plane is in all storage positions. Thus, the high-order position of a word can be placed at any storage address.

### 1.1.05 Scanning

The core-storage unit can only read out one character at a time. To read out a whole word, it is necessary to read out character by character. The order that these characters are read out is determined by the needs of the machine. For example, when two fields are added together, the units positions of the fields must be read out first to determine the signs of the fields. Information read out to an output machine is read out high order to low order.

The treatment of these words, character by character, is known as scanning. Reverse scanning is when the word is read out from high order to low order as in reading out to I/O machines. Forward scanning is the reading out of the low order position first.

## 1.2.00 STORED PROGRAMMING

The IBM 1410 system is controlled entirely by stored programming. A sequence or program routine of operations is stored or loaded in the core-storage unit before the system processes data. The processing unit then proceeds, step by step, through the stored-program routine, analyzing each instruction and performing the function called for before proceeding to the next instruction.

Just as program routines, or jobs in the machines using control panels can be changed by changing control panels, a stored-program routine can be changed by loading a new routine into the storage unit via some input machine.

### 1.2.01 Instruction Word

The instructions in core storage consist of a variable number of alphanumerical characters that combine to make up an instruction word. Valid instruction words vary in length from one to twelve characters, depending on the amount of information that the operation requires.

Valid instruction formats are:

- 
- d
- XXX d
- AAAAA
- AAAAA d
- XXX BBBBB d
- AAAAA BBBBB
- AAAAA BBBBB d

○ signifies the single-character operation code, AAAAA is the five-character address of the A field. BBBBB is the five-character address of the B field. XXX is, the I/O unit and control field, and d is the operation modifier.

The instruction words are arranged in core storage in sequence. Successive instructions are located in higher-numbered storage positions. Each instruction word must have a word mark with its operation code. It must not have any other word marks when executing the instruction, or a machine-stop results. There must also be a word mark in the location immediately to the right of the low-order character of the instruction word that ordinarily is the operation code for the next instruction.

The address of an instruction for execution purposes is the location of its operation code. No operation code has two valid lengths that differ by only one character. Instruction-length checking assures that the instruction length as read out of storage is one of the valid lengths for the particular operation code.

### 1.3.00 CONTROL

The central processing unit (CPU) is made up of four major areas: a storage unit, an address unit, an operational unit, and a control unit.

1. The storage unit is the center of all data flow in the IBM 1410. Information reads into storage from the assembly channel and reads out through a B-data register to the B-channel. A five-position address is fed to the storage unit, to control the position of storage to be used.
2. The address unit is used to store and to modify the five-position addresses that are used to locate the position of storage to be used.
3. The operational unit contains all the units necessary to edit, compare, move or add data.
4. The control unit contains registers, cycle controls, and a clock that are necessary to perform the various functions of the CPU.

The operation of the IBM 1410 system is controlled by the program routine stored in the core-storage unit. The completion of each program step requires two phases called instruction (I) phase and execute (E) phase.

### 1.3.01 Instruction Phase

I-phase is the first portion of the program step that is required to read the instruction word out of storage. The instruction word is made up of addresses, Op code, Op modifier, and control characters for I/O operations. As the characters read out during I-phase, they are stored in registers in the control or address units. A storage cycle is required to read each character out onto the B-channel. Because the length of the instruction words is variable, the number of cycles in I Phase is variable.

I-phase is further divided into I-cycles. To identify which character of the instruction word is reading out of storage during an I-cycle, an I-ring counts the characters or cycles. The I-ring consists of 13 triggers labelled I-ring Op, and I-ring 1 through I-ring 12.

As the first I-cycle of I-phase is taken, the I-ring is set to I-ring Op-time to identify this character as the Op code. The I-ring then advances to I-ring 1-time to identify the next character that reads out as either a modifier, or a ten-thousands position of an address, or the channel-select character, depending on the Op code character that has already been read out and recognized.

An example of I-phase is the data move instruction word which can consist of the Op code, followed by an A- and B-field address and a d-modifier character:

$\begin{matrix} \vee \\ \text{D} \end{matrix}$ 
AAAAA
BBBBB
d
 $\begin{matrix} \vee \\ \text{O} \end{matrix}$ 
AAAAA
BBBBB
d

At I-ring Op time, the Op code reads out of the storage unit and is set into the Op register in the control unit. The Op register is decoded to identify the next five characters as the A-field address.

I-rings 1 through 5 gate the characters into the proper positions of the A-address register that is located in the address unit.

I-rings 6 through 10 gate the next five characters to the B-address register in the address unit. The character read out at I-ring 11-time is the d-modifier. It is stored in the Op-modifier (Op-Mod) register in the control unit. During I-ring 12-time, the next character is read out to make sure that a word mark is located in the next position of storage.

#### I-Phase

I-cy	I-cy	I-cy	I-cy	I-cy	I-cy	I-cy	I-cy	I-cy	I-cy	I-cy	I-cy	I-cy
I Op	I 1	I 2	I 3	I 4	I 5	I 6	I 7	I 8	I 9	I 10	I 11	I 12
$\begin{matrix} \vee \\ \text{D} \end{matrix}$	A	A	A	A	A	B	B	B	B	B	d	$\begin{matrix} \vee \\ \text{O} \end{matrix}$

Op  
Reg.

AAR

BAR

Op  
Mod.  
Reg.

### 1.3.02 Execute Phase

At the completion of I-phase, the machine is ready to perform the actual operation. This portion of the program step is called execute phase (E-phase). The length and complexity of E-phase depends on the type of Op code. E-phase can consist of A, B, C, D, E, or F-cycles, with storage either forward or reverse-scanned. The length of each cycle varies from 4.5 to 7.5 microseconds. At the completion of E-phase the machine normally returns to I-phase to initiate the next instruction.

### 1.4.00 DATA FLOW

In processing data with the IBM 1410, three major operations are necessary:

1. Data must be fed into the IBM 1411 Central Processing Unit (CPU) from an input machine.
2. The CPU processes the data as controlled by the stored program.
3. When the processing is complete the result is fed to an output machine.

The center of data flow in the 1411 CPU is the core-storage unit that receives data from the IBM 1405 Disk Storage Unit, the IBM 1415 Console, or the IBM 1414 Input-Output (I/O) Synchronizer (Figure 1.4-1). When processing is complete, the data can be sent to the 1405, 1415, or 1414.

The 1414 I/O Synchronizer synchronizes and controls the various input-output units that are used in conjunction with the 1410 system. The 1414 includes the integrated synchronizer, print storage, and tape adapter units.

The integrated synchronizer is an input-output control unit that contains up to eight 80-character magnetic-core registers. These registers provide independent buffering for all associated I/O units. A data transfer between the CPU and the card read-punch or paper tape reader takes place through the integrated synchronizer. Processing time is saved because the CPU does not have to wait for a mechanical unit to pick up speed, etc.

The print storage unit contains a 132-position core-storage unit that is an intermediate storage device between the CPU and the IBM 1403 Printer. Included in the Print Storage is the circuitry necessary to control the carriage for forms handling.

The tape adapter units provide a control system for the various tape units that may be used with the 1410 System.

### 1.4.01 Central Processing Unit Data Flow

Every operation in the CPU requires a storage cycle. At the beginning of every storage cycle a five-position address is set in the storage address register (STAR), where the output is decoded to select the desired position of core storage (Figure 1.4-2). Depending on the size of storage, two or four characters are read out of core storage into the B-data registers. The ten-thousand position of STAR selects which register is to read out on the B channel. If the storage cycle is taken only to

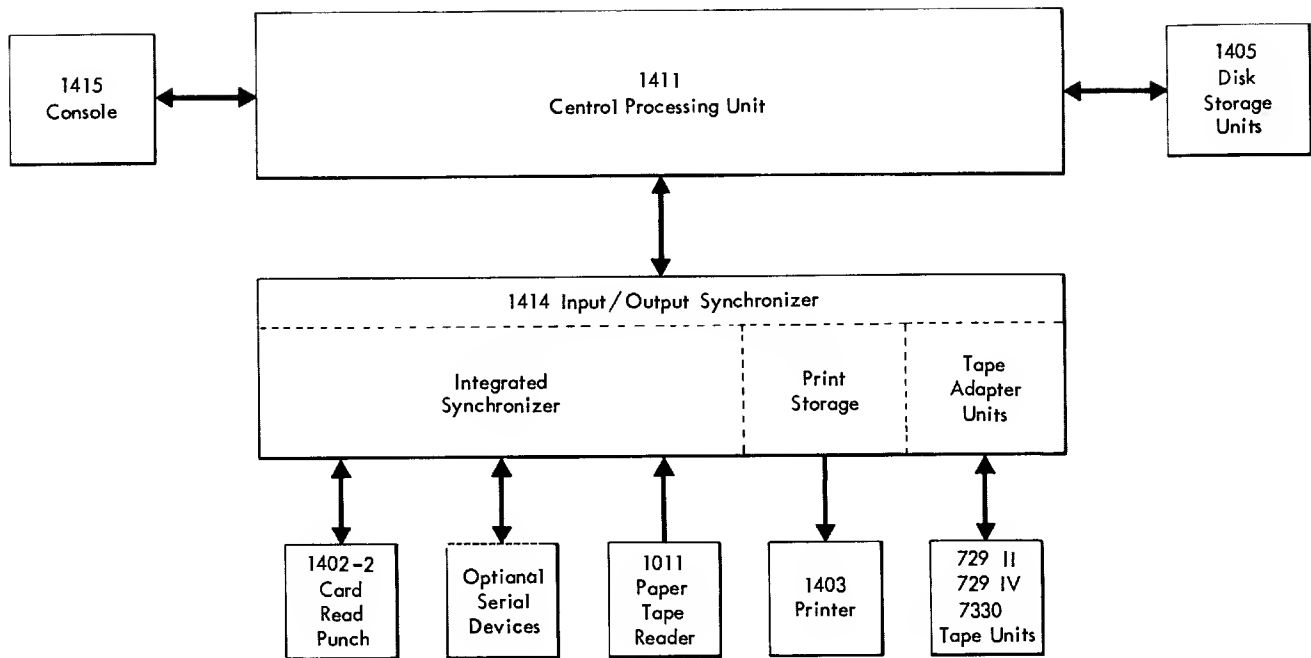
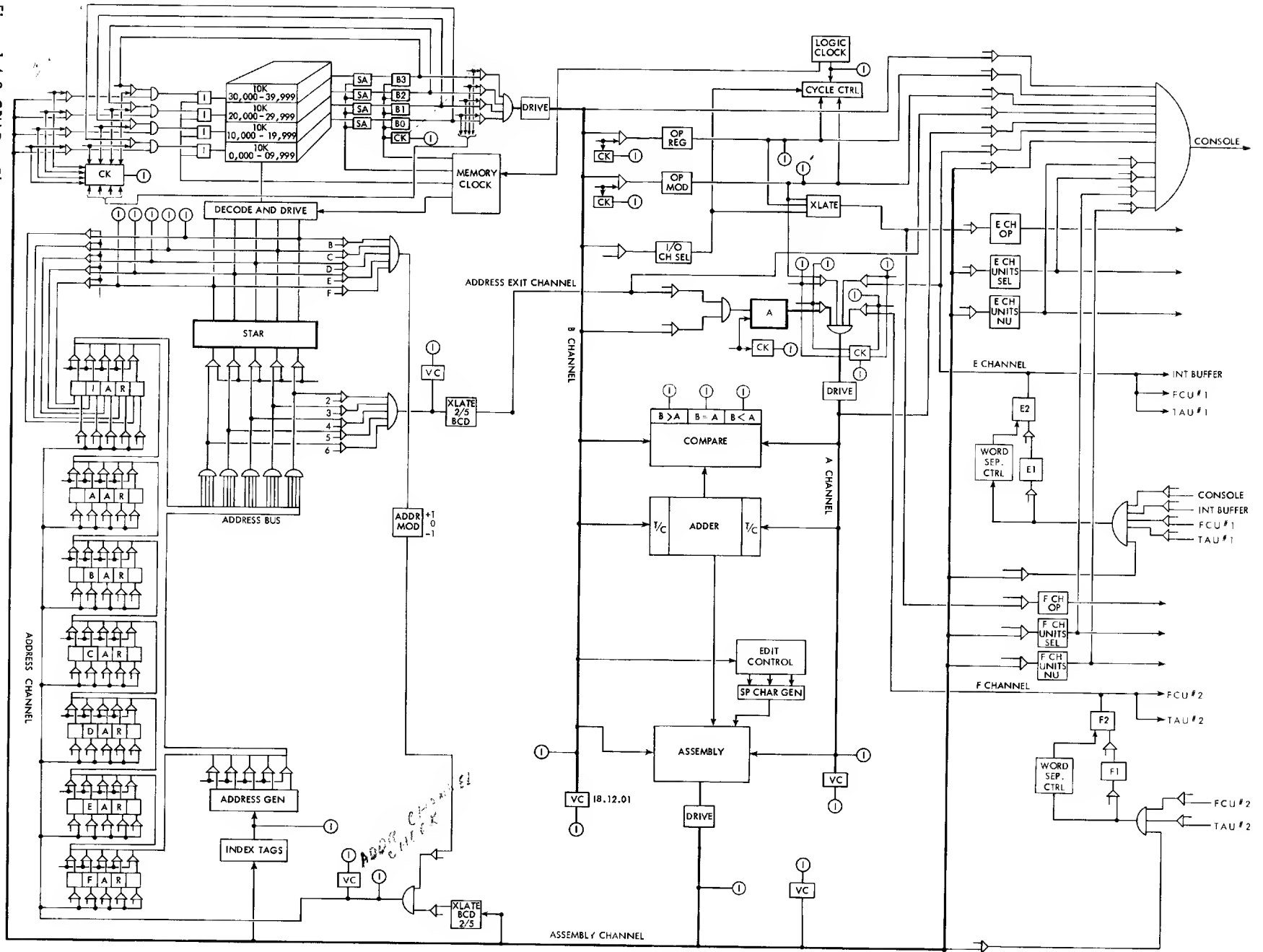


Figure 1.4-1 Data Flow

Figure 1.4-2 CPU Data Flow



read out a character, all four B-data registers are read back into storage (regenerated). If the storage cycle is taken to write a new character into storage, the B-data registers that are not selected are gated back into storage, while the selected character is read in from the assembly channel.

When the character is on the B-channel, it is available to most of the units throughout the machine. For example, when a character is read out during an I-cycle at I ring Op-time, it is gated from the B-channel to the Op register. At Op-modifier character time, the character is gated to the Op-mod register.

During E-phase, if the character is a B-field character, it can be gated to the B-side of the compare matrix, of the adder, of the edit unit, or of the assembly unit. If the character is an A-field character, it is usually stored in the A-data register where it can be gated to the A-channel to be available to the A side of the compare matrix, of the adder, and of the assembly unit. The character is stored in the A-data register until the corresponding character of the B-field is available on the B-channel so that the two characters can be compared or added. The assembly area combines the desired portions of characters from the A-channel, the B-channel, the adder, and/or special character generator, and gates the character that results to either the core-storage unit or an output machine.

For example, when two characters are added together in the adder, only the digit portions of the characters are combined. Any zones or word marks that appear over the B-field character are read back into storage unaltered. The assembly unit combines the adder output with the B-channel zones and word mark, and gates the new B-field character back to storage.

During I-phase every instruction word character that is read out of storage is gated from the B-channel to the A-data register, then to the A-channel. The assembly gates the A-channel zones, numerical bits, and the word mark onto the assembly channel. When the control unit identifies a character as an address, the assembly channel is gated to the address channel, where the character is set into the desired position of an address register.

For input operations, characters are gated into the E- or F-data registers from the input machine. The character on the E- or F-channel is gated to core storage via the A-channel, and the assembly unit. For output operations, characters that come from storage on the B-channel are gated through the assembly to set the E- or F-data registers. The E- or F-channel is gated to the output machine that is selected by the instruction word.



## 2.0.00 CENTRAL PROCESSING UNIT

### 2.1.00 CORE STORAGE

Magnetic ferrite cores are used as the high-speed storage medium because of their low access time. The core-storage unit has a basic cycle of 4.5 microseconds. This cycle is divided into read and write time. Read time makes a character that is contained in the storage unit available to the CPU. Write time allows characters to be entered into storage from other units. All information in storage is in BCD form. It is checked for parity upon entering or leaving storage.

#### 2.1.01 Ferrite Cores

Information storage is achieved by using the magnetic properties of the ferrite core. Each core can store a single bit of information, either a 1 or a 0.

##### Physical Properties

A ferrite core is a ceramic material with ferromagnetic properties. It is prepared from oxides of iron and other elements, particularly zinc, manganese and nickel. The materials are ground to a fine powder, carefully mixed together with a binder and then pressed into shape. After baking, the core is hard and brittle. It is in the form of a short, hollow cylinder that measures 50 mils outside diameter, 30 mils inside diameter, and 12 mils height.

##### Magnetic Properties

The ferromagnetic properties of the core permit it to be magnetized by applying an external magnetizing force. It can be magnetized in either of two directions, north-to-south, or south-to-north. After the core is magnetized in either direction, it remains magnetized until another force is applied that is enough to change the direction. The core is stable in either of these two directions and is called a bistable storage device.

The magnetic force necessary to flip (change the direction of) a core is provided by the magnetic field that surrounds a wire with current passing through it. This wire passes through the hole in the core. When the magnetic field is great enough in a direction opposite to that of the core, the core flips. If a core is set in one direction, and the current through the wire produces a field to set the core in the same direction, a very small change takes place in the core. If a core is set in one direction, and the current through the wire produces a field to set the core in the other direction, and the field is not large enough, the state of the core is not changed significantly. The amount of current necessary to flip a core is called full-select current. When a core flips, there is a rapid breakdown and buildup of the magnetic field around a core. This changing field induces a voltage in another wire in the core. This sets a latch and indicates that the core is set. This wire is called the sense winding.

#### 2.1.02 Core Plane

To make use of a large number of cores, it is necessary to control which cores

are set or not set. Consider two parallel wires going through a core, each carrying half-select current in the same direction. The resulting magnetic force from each current is additive, and equals the effect of full-select current. Now, shift one wire 90 degrees. With the same half-select current flowing, the effect is still additive, and produces enough field to switch the core. The wires are called X and Y drive lines. Figure 2.1-1 shows a core plane. To interrogate the white core in Figure 2.1-1, impulse the appropriate X- and Y-drive lines with half-select current. Only the white core experiences the effects of both half currents. If the core contains a 1, it is switched, and a pulse is induced in the sense winding. This pulse is amplified and is used to set a latch. If the addressed core contains a zero, there is no pulse to set a latch. Note that the state of flux in the other cores along the two drive lines is not changed by the half-select current. Therefore, they induce no pulses in the sense winding.

### 2.1.03 Core Array

There are eight cores used to store one character in BCD form. Each core stands for one of the eight bits (1, 2, 4, 8, A, B, C, WM).

The core array has 8 planes corresponding to the 8 bits. Each plane has a sense winding that goes through every core on that plane (Figure 2.1-2). The number of cores on each plane determines the number of characters that can be stored.

#### Read

To read a character out of storage, the appropriate X- and Y-lines are each impulsed with half-select current (called read current). Every core that is set at the intersection of these two lines is reset. The pulse is sensed on the corresponding sense winding where it is amplified and used to set a data register latch.

#### Write

To write into a position, the same X- and Y-drive lines (as in read) are used to select the proper location. However, to write, the current through the lines is in the opposite direction (called write current). Because both of these are half-select currents, they would, unless modified, set all cores in a position. To control which cores are set, another wire (inhibit line) is passed through every core on a plane (Figure 2.1-2). To prevent setting a core, half-select current is fed on the inhibit line in a direction opposite to the X- and Y-currents. This cancels out half the magnetic force created by the X- and Y-lines and only half-select current passes through a core. In setting a core, no current is applied to the inhibit winding. This permits full-select current to set the core.

#### Physical Arrangement

The physical arrangement of the cores and the X- and Y-, inhibit, and sense windings of a plane are shown in Figure 2.1-3.

Each X- and Y-drive line goes only through the cores in one row or column. The inhibit winding goes through all cores in an electrical plane, and parallels the X-drive lines. The sense winding is threaded in parallel with the Y-drive lines,

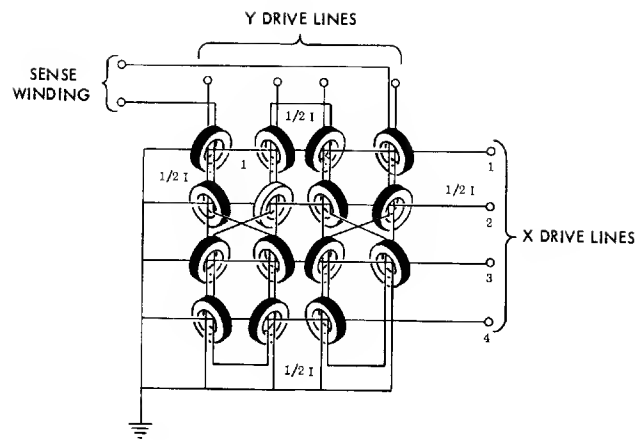


Figure 2.1-1 Core Plane Winding

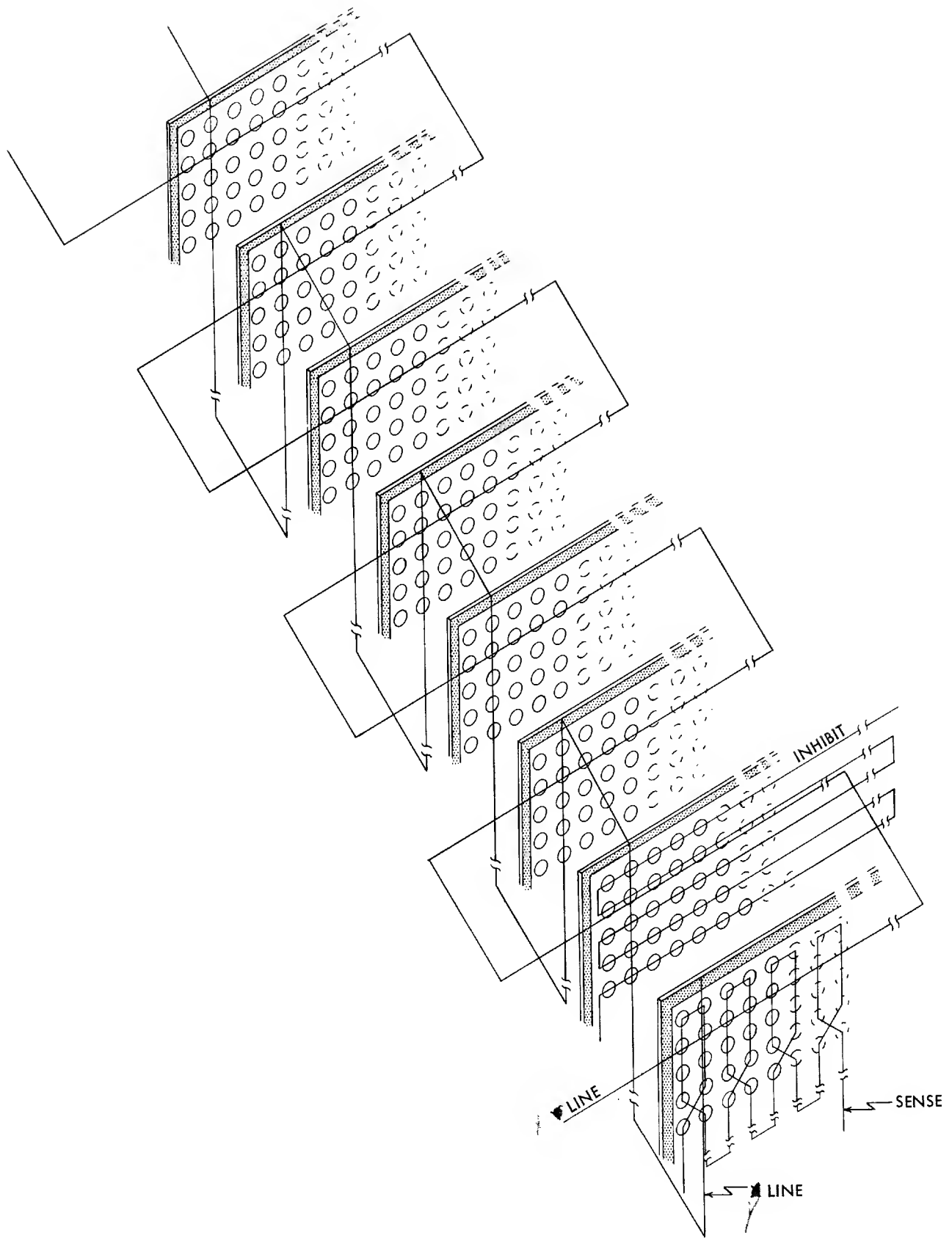


Figure 2, 1-2 Core Array Winding

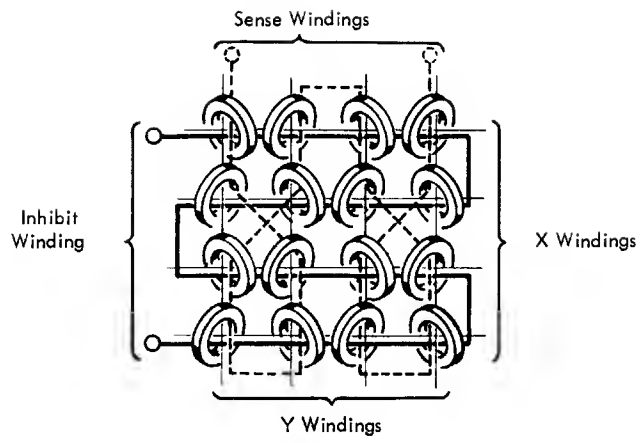


Figure 2.1-3 Core Plane Section

and goes through all cores in an electrical plane. Also the sense winding crosses over at the middle of the plane to cancel the effect of mutual coupling between the lines. In the 1411 core storage, there are actually two inhibit and two sense windings per plane. Thus, for the inhibit and sense functions, the 100-by-100 array is actually two 50-by-100 core planes.

The essential concern is the voltage that is induced in the sense winding as the result of the flux change in the addressed core that is set. The sense winding is crossed at the midpoint of the plane so the Y-drive pulse does not induce enough voltage to be interpreted as a 1, even though the core contains a 0.

For simplicity, Figure 2.1-4 shows the sense winding in electrical plane 1 and the inhibit winding in electrical plane 2. The sense winding of electrical plane 2 is a mirror image of the sense winding in electrical plane 1. The inhibit winding in electrical plane 2 is a mirror image of the inhibit winding in electrical plane 1.

#### 2.1.04 Data Flow Components (Figure 2.1-5)

The data-flow components are:

1. data-register latches
2. inhibit drivers and sense amplifiers
3. logical switching networks.

Because the induced-voltage pulse on the sense line is of low amplitude, a sense amplifier amplifies the pulse to the logic circuit level. It is then used to set a data register latch.

To select the induced pulse from the noise impulses on the sense winding, a short pulse (strobe) gates the sense amplifiers just as the ferrite core reads out.

The data-register latch holds information that is to be again written into the cores. This rewriting of information back into the cores is called regeneration. The time that regeneration occurs is controlled by the inhibit gate and a control line called Storage Regen. This specifies that original information is to be written back into the cores. The data register latch causes the inhibit drivers either to suppress or to allow half-currents to flow in a direction opposing the write currents. Because the write current is in a direction that is opposite to the read current, all cores of an address are set to 1 during the write cycle. The inhibit driver prevents the writing of a 1 by an opposing half current that is under the control of every data-register bit-position that contains a 0.

#### Storage Cycles

A core-storage unit has two basic types of cycles.

A read-out cycle occurs when information reads out of the ferrite cores and passes on to other units of the system. The information read-out is regenerated (written back into the same cores) during the write portion of the operation.

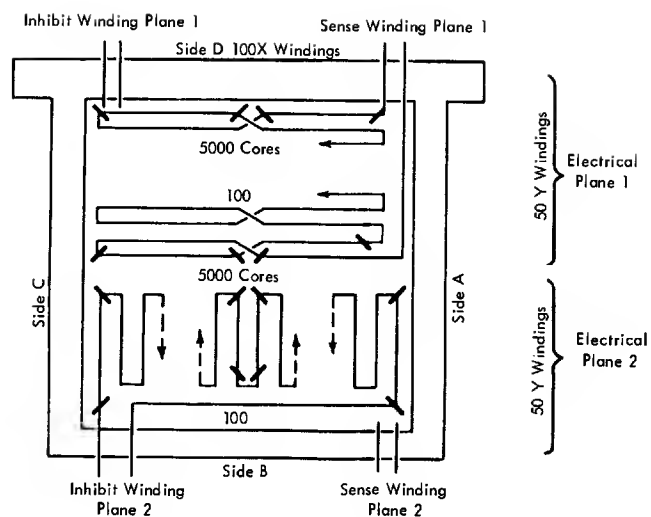


Figure 2.1-4 Actual Core Plane

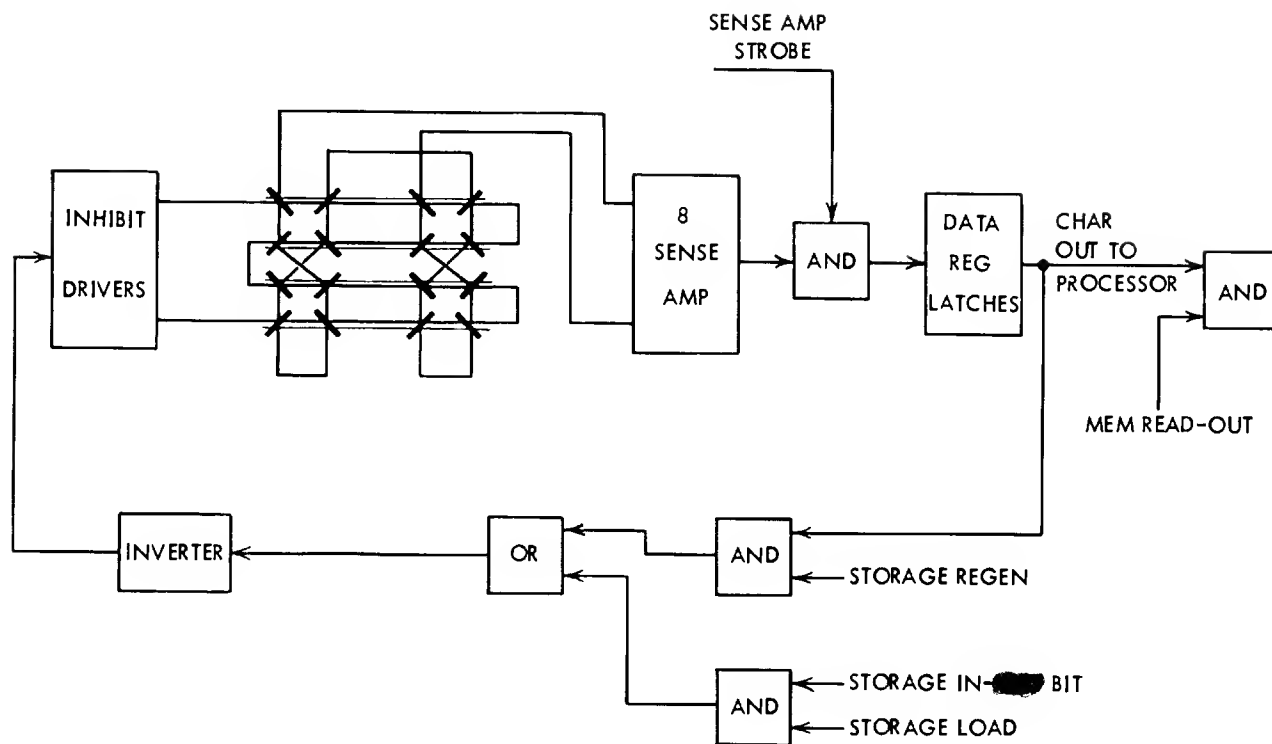


Figure 2.1-5 Core Storage Data Flow

A read-in cycle occurs when new information from another unit of the system is to be written into the cores.

The information that is already in the cores reads out to the data register the same as during the read cycle. The write portion of the cycle then writes the new information into the cores. Both read-out and read-in cycles are discussed in the sections entitled Read Out and Read In.

#### Read Out

To read out of core storage, these steps are necessary:

1. Read the cores in order to set the information in the data latches.
2. Write the information from the data latches back into the cores.

Read. During the first half of any machine cycle (Figure 2.1-5) that uses core storage, the same relative core in each plane is pulsed with X- and Y-current in the read direction. This sets all cores at a given address to 0. Any 1's present are picked up on the sense lines, amplified by the sense amplifiers, and used to set the data latches. Cores that contain 0's do not activate any sense lines. The latches associated with these cores remain at 0.

Write. During the last part of any storage cycle (Figure 2.1-5) the same X- and Y-drive lines are always impulsed in the opposite, or write, direction. This current pulse is in the direction to place a 1 in the addressed cores. However, if the data-register latches are OFF (0), the inhibit gate pulses the inhibit winding during write time. Because this inhibit pulse is in the direction opposed to the write pulse in the X-winding, the addressed core is pulsed with half-select current that leaves it in the 0-state.

#### Read-In

To read new information into core storage, three steps are necessary:

1. Read the cores in order to gate information that is already in the cores into the data register.
2. Activate the inhibit drivers (Figure 2.1-5) with the new information from some other unit of the system.
3. Write the new information into the cores during the write cycle.

Read. As with every storage cycle, an X- and Y-drive line is pulsed with read current. The sense amplifiers are strobed and the information in the cores is gated to the data latches. The read portion of the cycle thus sets all cores in the addressed location to 0, and also sends the information that was in the cores to the data register.

The new information to be written into storage is gated into the inhibit driver by the storage load signal.



The rest of the storage read-in cycle is the same as the write cycle of a read-out operation.

### 2.1.05 Matrix Switching

To read out a position of storage, the proper X- and Y-lines must be addressed (selected). The device used to address these lines is called a matrix switch. Matrix switches are used to simplify X- and Y-addressing. The outputs of the matrix switch drive the X- and Y-lines that drive the ferrite cores. The matrix switch makes it possible to address any one of 100 X-drive lines (in a 10K core array) with 16 drive circuits plus one of 10 switch-selector gates. The determination of 1 of 100 Y-drive lines proceeds in exactly the same way. Both reading X-Y drive currents and reversed writing X-Y drive currents are obtainable by the matrix switch.

#### Physical Properties

A matrix switch has ten special pulse transformers that are housed in rectangular-shaped, heat-dissipating plastic (Figure 2.1-6).

The core of each transformer is made up of 12 rings of powdered iron, stacked in two parallel 6-ring cylinders. Each transformer has 16 primary windings and one secondary winding. The 16 primary windings are interconnected, with end windings that are connected to matrix-switch drivers. Secondary windings are connected to the X- or Y-drive windings.

#### Magnetic Properties

~~The core material is similar in magnetic properties to the ferrite cores. There is a certain amount of current necessary to operate the switch, and any less current has little or no effect on the switch.~~

#### Matrix-Switch Operation

Matrix-switch operation is made possible by the addition and cancellation of fluxes from eight primary winding currents.

For simplicity, consider a matrix switch with two outputs and four inputs (Figure 2.1-7). With both cores reset, if current passes in the same direction in lines 1 and 2, the flux from each line adds at core 1 to set the core. This causes an impulse to be induced in the output winding. As the currents pass through core 2 in opposite directions, there is a cancellation of flux, thus no output. When lines 3 and 4 are impulsed in that same direction, core 1 is reset. This induces an impulse into the output winding in the opposite direction. Again there is no output from core 2. In core 1, lines 1 and 2 are used for forward (read) output. Lines 3 and 4 are used for a reverse (write) output. In considering core 2, lines 1 and 3 are used for a read output; and lines 2 and 4, for a write output.

Now consider a ten-position matrix switch. The matrix switch requires eight simultaneous currents for an output. There are 16 input lines as illustrated in Figure 2.1-8. This shows the input lines required to obtain a read and write out-

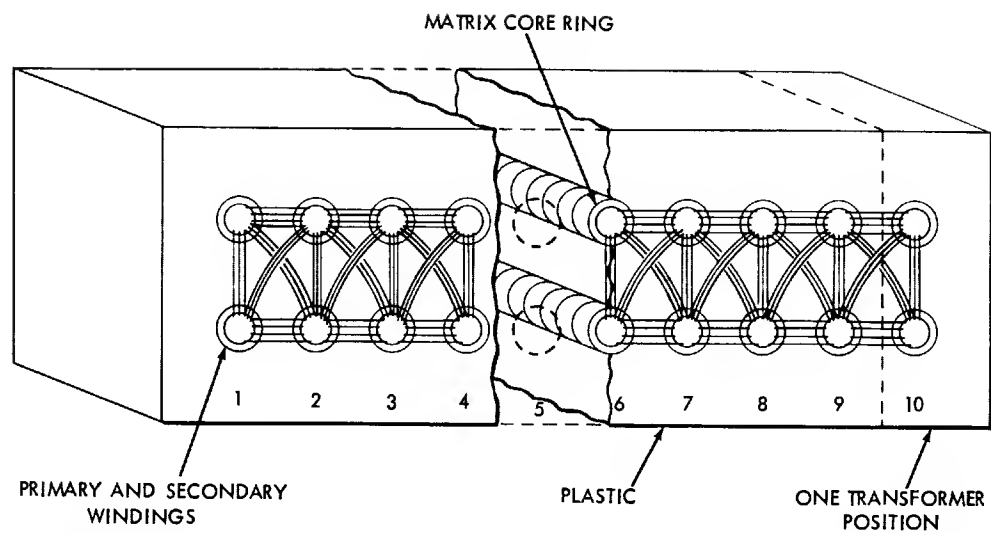
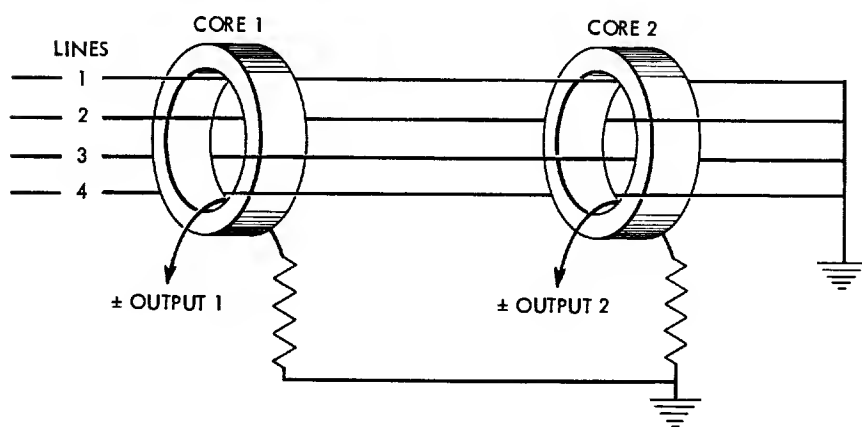


Figure 2.1-6 Matrix Switch



READ (FORWARD)

MATRIX CORE OUTPUT	ACTIVE LINES			
	1	2	3	4
1	X	X		
2	X		X	

WRITE (REVERSE)

MATRIX CORE OUTPUT	ACTIVE LINES			
	1	2	3	4
1			X	X
2		X		X

Figure 2.1-7 2 Position-Matrix Switch and Truth Table

MATRIX CORE OUTPUTS	ACTIVE LINES DURING															
	READ = +								WRITE = 0							
	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
0	+	0	+	0	0	+	0	+	0	0	+	+	+	0	+	0
1	0	0	0	+	+	+	0	+	+	+	0	+	+	0	0	0
2	0	0	+	0	+	+	+	0	+	0	+	+	0	+	0	0
3	0	0	+	+	0	0	+	+	0	+	+	0	+	+	0	0
4	0	+	0	0	+	0	+	+	0	+	+	+	0	0	+	0
5	0	+	0	+	0	+	+	0	+	0	+	0	+	0	+	0
6	0	+	+	0	0	+	0	+	+	+	0	0	0	+	+	0
7	+	+	0	0	0	0	0	0	+	+	+	+	+	+	0	0
8	+	0	0	0	+	0	+	+	+	0	0	0	+	+	+	0
9	+	0	0	+	0	+	+	0	0	+	0	+	0	+	+	0

Figure 2.1-8 Truth Table for 10-Position Matrix Switch

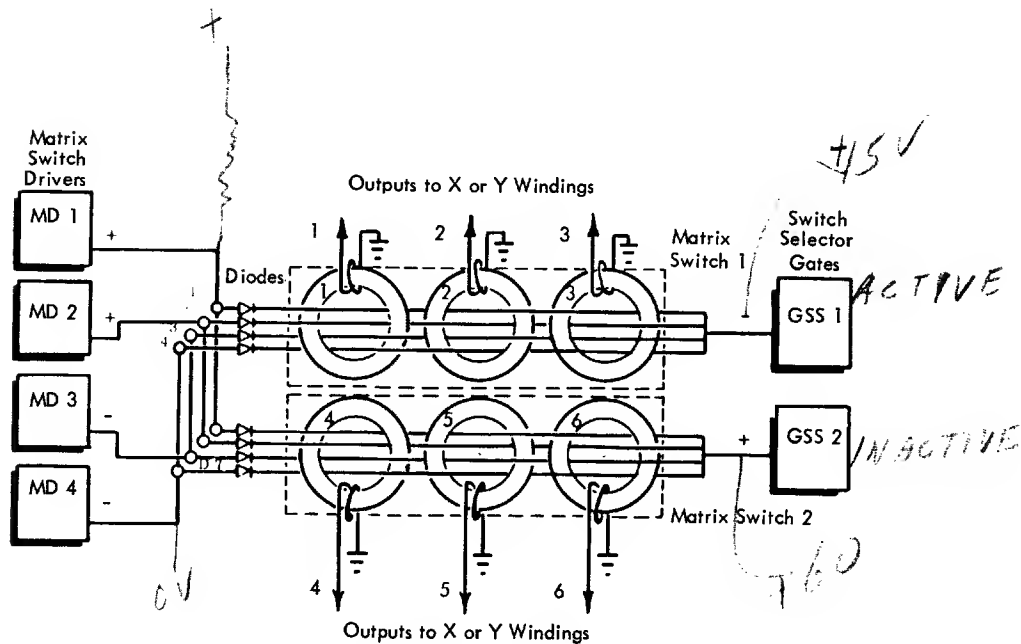


Figure 2.1-9 Matrix Switch Drivers and Gates

put for a ten-position matrix switch. In core 1, for example, lines 1, 3, 6, 8, 11, 12, 13 and 15 must be conducting in order to give a read (forward) output; lines 2, 4, 5, 7, 9, 10, 14 and 16 must be conducting to give a write (reverse) output. Note: during a basic memory cycle, the read currents come during the first half of the memory cycle, and the write currents come in the latter half of the memory cycle. Also, by combinations of selected input lines, any one of the ten outputs can be selected. To use matrix switch selection fully, it is necessary to operate several matrix switches in parallel and to select the output of the desired matrix switch. Figure 2.1-9 shows two 3-position matrix switches operating in parallel with switch selection.

In Figure 2.1-9, if matrix-switch drivers 1 and 2 are positive and matrix switch drivers 3 and 4 are negative, cores 1 and 4 are conditioned. Because switch selector gate 1 is negative, adding currents flow to matrix-switch drivers 1 and 2 from switch-selector gate 1. Cores 2 and 3 have cancelling currents flowing. Cores 4, 5 and 6 have no current flowing to the positive matrix-switch drivers 1 and 2, because switch selector gate 2 is positive. No current flows from negative matrix-switch drivers 3 and 4 through cores 1, 2 and 3, because switch selector gate 1 is negative. Diodes 7 and 8 prevent reverse currents from flowing through cores 4, 5 and 6 from matrix-switch drivers 3 and 4 to gate 2. The result is that only core 1 has the necessary currents to produce a read output. A reverse (write) output is obtained if matrix-switch drivers 3 and 4 are made positive and matrix switch drivers 1 and 2 are made negative, and no change is made on the switch-selector gates.

A truth table for read currents appears in Figure 2.1-10.

By reversing the plus and minus signs under the heading of Matrix-Switch Drivers Only (Figure 2.1-10), the truth table for writing is obtained.

The IBM 1410 uses ten 10-position matrix switches and ten switch-selector gates to select one of 100 X-drive lines. Ten matrix switches and ten switch-selector gates provide Y-line selection on a 20K or 40K machine. A 10K machine uses only five 10-position matrix switches with five switch-selector gates.

#### 2.1.06 Noise Problems

Unwanted noise from many sources occurs throughout any core-storage unit. The larger the unit, the worse the noise problem becomes.

There is an unavoidable amount of mutual coupling between parallel lines, even with careful spacing and shielding. The cores themselves provide another major source of noise. No core remains completely saturated and the half-select current tends to move the flux state slightly. The change of the flux state is either toward or away from saturation.

Noise in the sense winding occurs when a core is on the impulsed drive line but is not at the intersection of two drive lines. If all the induced voltages from the half-selected cores were added together, the effect would be troublesome.

One feature that reduces the half-select noise problem is the staggering of read pulses. By energizing one drive line (Y-drive line) in advance of the other (X-

Outputs to X or Y Lines	Matrix Sw Drivers				Sw Sel Gates	
	1	2	3	4	1	2
1	+	+	-	-	-	+
2	-	+	-	+	-	+
3	+	-	-	+	-	+
4	+	+	-	-	+	-
5	-	+	-	+	+	-
6	+	-	-	+	+	-

Figure 2.1-10 Truth Table for Two  
Three-Position Matrix Switches

drive line), the number of half-selected cores at a given time is reduced. This reduces noise. Later in the cycle, when the X-drive line energizes, and the sense-winding output is used, the noise from the half-selected cores on the earlier Y-drive line has subsided. This delayed read pulse reduces the noise problem.

A second method of noise reduction results from physical arrangement of the windings of the core plane. Cores, drive windings, and sense windings are so arranged that one-half of the half-selected cores tend to cancel the noise induced by the other half of the half-selected cores of a plane.

#### 2.1.07 1411 Core-Storage Unit

The 1411 core storage unit reads in or reads out a single character during a 4.5-microsecond cycle. Reliability is insured by parity checking all information read into or out of the core-storage unit.

#### 2.1.08 Core-Storage Addressing

The 40K core array requires 40,000 valid five-position addresses from 0,000 to 39,999. The address is read from the CPU into the storage address register (STAR). The STAR stores the address while its output is encoded in order to select the proper character to read out.

#### Operation

Figure 2.1-11 shows the locations of the addresses in storage and the encoding that is necessary to select the X- and Y-lines. When the address is read into the STAR, the units and tens positions are encoded to drive and to gate the ten load-sharing matrix switches (LSMS) that select the X-line. The hundreds and thousands positions of STAR are encoded to drive and to gate the ten matrix switches that select the Y-line. The coincidence of these two lines causes four characters to be read out of cores into the four B-data registers. The ten-thousands position of STAR is fed to the character-selection circuits that determine the character to be addressed. The output of character selection is fed to Character Regen or Load where it combines with either Memory Regen (read operations) or Memory Load (write operations) to control which characters are regenerated, read out of storage, or loaded into cores from the assembly area. If Memory Regen is up, all character regen lines are brought up to reread the characters into cores. Also a character-select line is brought up to gate the desired B-data register onto the B-channel. When Memory Load is up the character that is position-selected is loaded from the assembly area, while the other three characters are regenerated.

#### 10K Core Array

Figure 2.1-12 shows the locations of the addresses in a 10K storage unit and the encoding that is required to select the proper character. When the address is read into STAR, the units and tens position of STAR are encoded to drive and to gate the ten matrix switches which select the X-line. The hundreds and thousands positions of STAR select the Y-line. The coincidence of the X- and Y-lines causes two characters to be read out into the two B-data registers. The thousands posi-

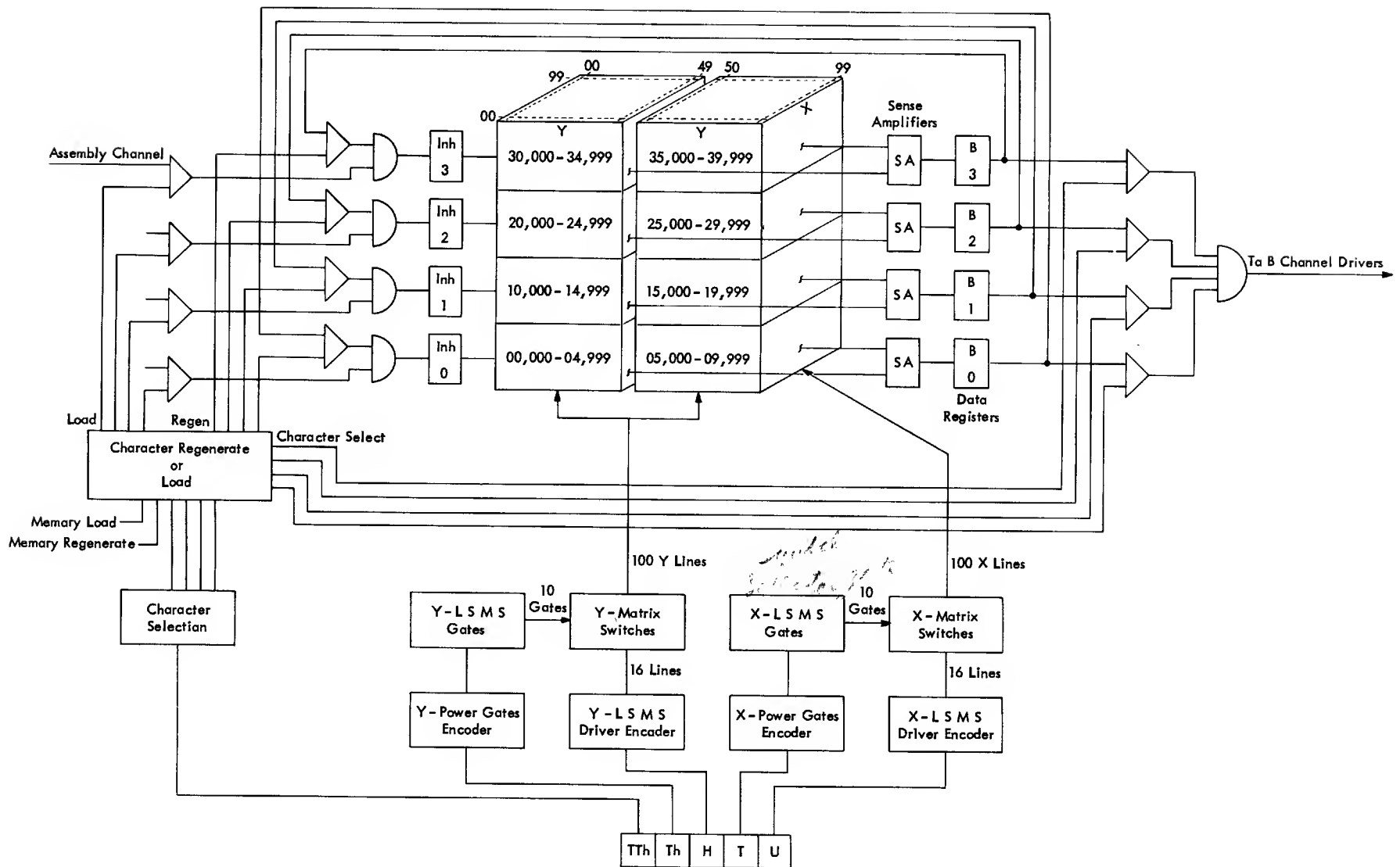


Figure 2.1-11 40K Core Storage Addressing

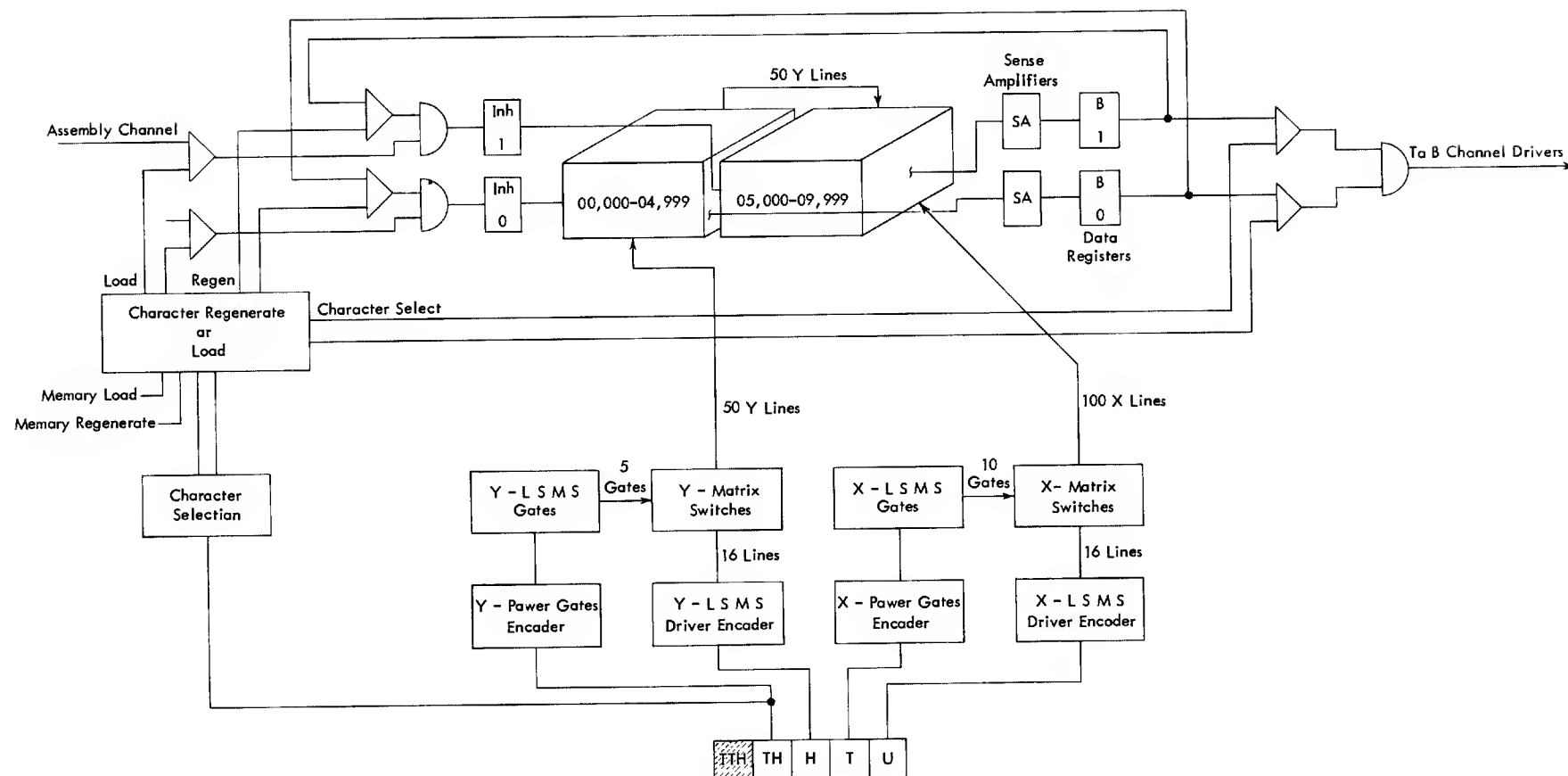


Figure 2.1-12 10K Core Storage Addressing



tion of STAR is fed to the character select where it combines with Load Memory or Memory Regen to gate the character into or out of storage.

### 2.1.09 Memory Clock

Every storage cycle, regardless of CPU operation, has a read and write time. On a read-out operation, the write time is used to regenerate the data into the cores. On a write operation, the cores are read out to clear the position where the new character is to be read in.

The memory clock provides the read and write pulses that are required by the storage unit.

#### Read

The CPU initiates a storage cycle with a read call that, along with Start Memory Clock, resets the B-data registers, activates the X- and Y-power-gate controls, and initiates a .425  $\mu$ s delay (Figure 2.1-13).

The X- and Y-power-gate control lines combine with the output of the X- and Y-power gates encoder to drive the X- and Y-LSMS gates. After the .425  $\mu$ s delay, the Y Rd line gates the hundreds position of STAR to the Y-matrix-switch driver encoder that, along with the matrix-switch gates, drives the selected Y-line.

Y Rd also activates a .280  $\mu$ s delay that initiates the X Rd to gate the X-line-drive.

The X Rd line impulses two more delays. One delay, of .725  $\mu$ s, initiates Strobe that controls the sense amplifiers. The other delay of 1.100  $\mu$ s, drops the X Rd and Y Rd lines to complete the read-out of cores.

Strobe also initiates a .350  $\mu$ s delay to drop itself. Figure 2.1-14 is a timing chart for a storage cycle. The delays include time lost through transistor delays. The dotted lines indicate the worst possible conditions.

#### Write

A write call from the CPU initiates the write portion of the storage cycle. When the memory clock receives a write call, it combines with Start Memory Clock to impulse a .15  $\mu$ s delay. The .15  $\mu$ s delay initiates a Z-pulse that gates the inhibit drivers. The .15  $\mu$ s delay also brings up Y Wr. This again gates the hundreds position of STAR to the Y-matrix-switch encoder to control the selection of the proper Y-line. Y Wr also impulses a .280  $\mu$ s delay that initiates X Wr to select the proper X- line.

All the conditions are now present either to regenerate or to write a new character into the cores.

X Wr also impulses a 1.2  $\mu$ s delay which drops the X- and Y-write lines. When Y-write drops, it turns off the Z-pulse. This in turn drops the X- and Y-power gate controls.

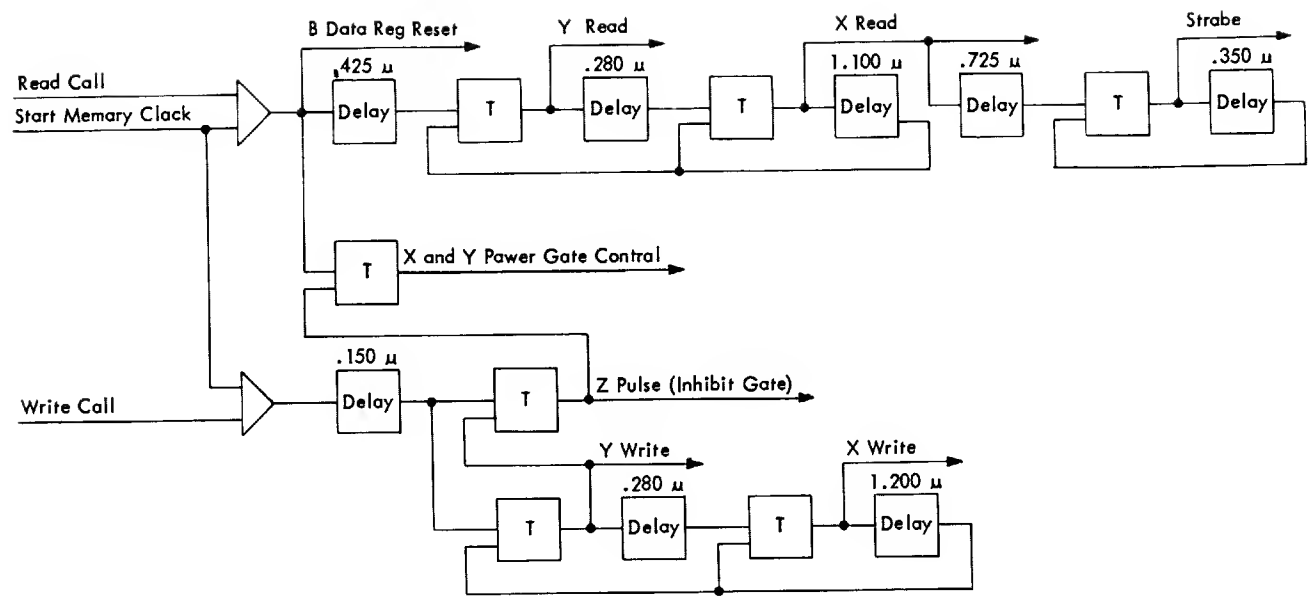
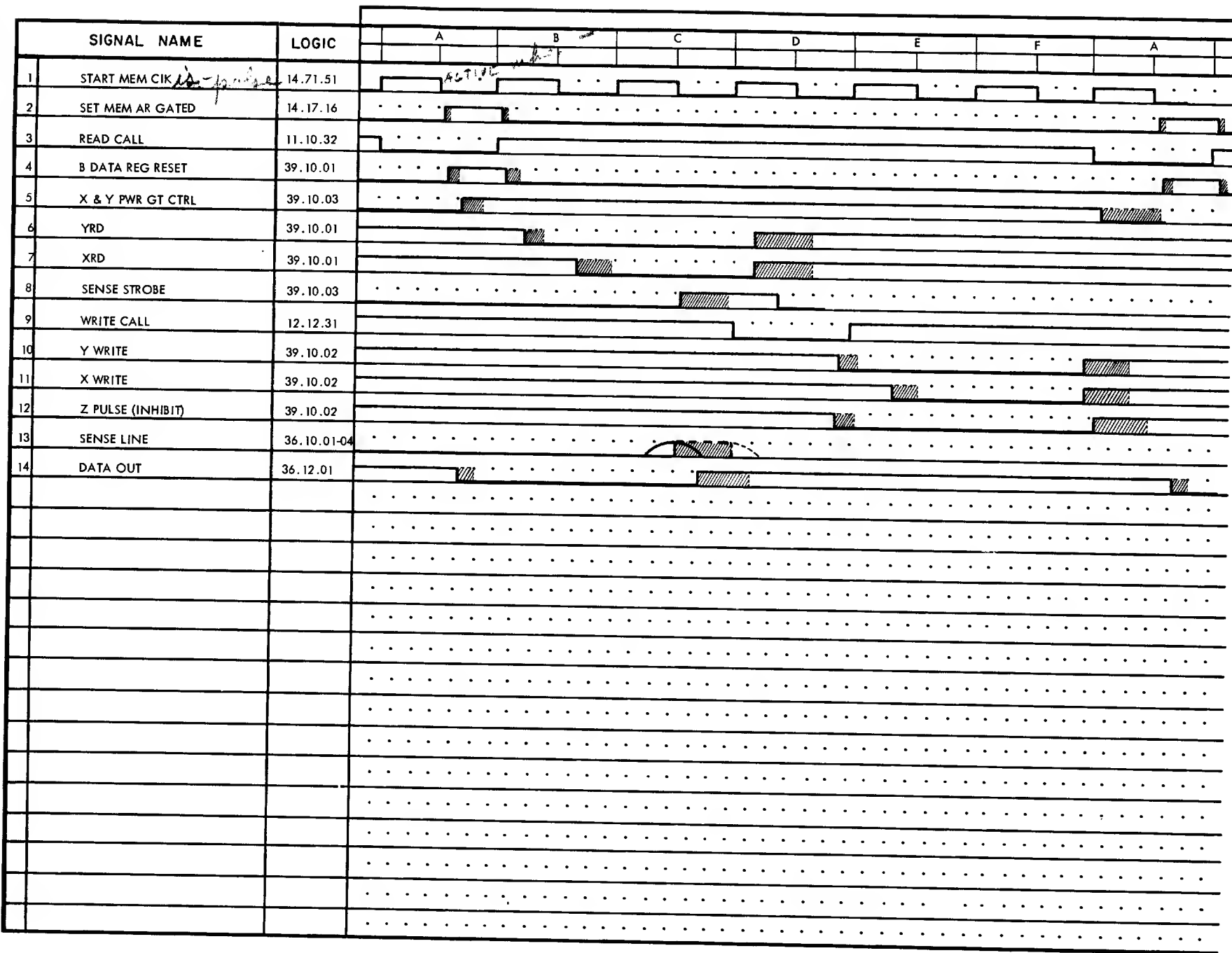


Figure 2.1-13 Storage Clock

Figure 2.1-14 Memory Clock



## Length of Storage Cycles

A storage cycle in which all characters are to be regenerated can be completed in  $4.5\text{ }\mu\text{s}$  or six logic-gate pulses. In this case, the write portion of the storage cycle starts at logic-gate D with the write call from the CPU.

On operations where a character is read out of storage to be combined with another character before it is read back into cores, the storage cycle is increased in length in order to allow the CPU time to construct the new character. In this case the write call is delayed to a later logic gate pulse.

## B-Data Register Set Check

The B-data register set check insures that the correct core-storage operation occurs. The circuit makes certain that on every storage cycle the B-data register receives an impulse to reset it.

Some of the failures this check recognizes are:

1. Failure to start the storage clock.
  - a. If the memory clock does not start, the last character that was read out of the B-data register again reads out. Without this check, the error would not be caught. It is caught because the memory clock must start to develop the reset pulse.
2. Failure to receive X- or Y-line current or strobe pulse.
  - a. If the B-data register is reset and no new data is read in, the B-channel validity-check circuit detects the error.

The circuit for the check is on logic 18.14.06. Both triggers are reset OFF. One trigger changes status on the fifth logic gate pulse of every storage cycle. The other trigger changes status on every B-data register reset pulse. As long as both triggers remain in the same condition (ON or OFF), no error is indicated. If the one trigger fails to receive a B-data register reset pulse, the triggers become opposite in status. The outputs of the triggers then cause a B-data register reset error. This stops the machine and lights the B-data register set-check indicator.

### 2.1.10 Character Gating

On a 20K or 40K core-storage unit, the ten-thousands position of STAR is fed to character selection (35.10.01) where the bit configuration brings up either RO character 0, 1, 2, or 3. RO character 0, 1, 2, or 3 combines with load or regenerate memory to select, or load that character (35.10.02). All characters are controlled to regenerate except a character that is to be loaded from the assembly area.

On a 10K machine, the thousands position of STAR is fed to character selection where the bit configuration for characters 0 through 4 brings up RO character 0, and 5 through 9 bring up RO character 1.

## B-Character Select Check

The addition of a B-character select check circuit increases the reliability of the core-storage unit. This circuit insures that one, and only one, character is either gated out to the B-channel or read in from the assembly unit.

### Operation

On a 40K storage unit there are eight character-control lines: four character-select, and four character-load. These eight lines combine (35.10.03) to bring up one of the two character-select error-check lines. If more than one character-control line is up, both of these character-select error check-lines are brought up to indicate an error condition. If none of the character control lines are up, both of the character-select error-check lines are down. This is also recognized as an error condition.

This operation check detects errors in programming that cause invalid addresses. Examples: 40,000-and-above on a 40K core array; 20,000-and-above on a 20K core array. On a 10K core array, invalid addresses are detected by the address-bus validity check.

## 2.2.00 ADDRESS UNIT

The address unit is used to store and modify the five-position addresses that locate the position of storage to be used. The address unit contains eight address registers. One of these, the storage address register (STAR) is used as the output of the address unit that is fed to the core-storage unit. The STAR is read into either from one of the seven other address registers (labeled A through F and I), or from the address generator (used for indexing). The address unit also contains an address modifier that is used to add one or subtract one from the address in STAR in order to obtain the address of the next position of storage.

### 2.2.01 Address Registers

An address register (AR) is a storage unit capable of reading in or out the five-digit addresses that are required to locate information in core storage. Characters are stored in the address registers in the two-out-of-five code.

The two-out-of-five code is a strictly numerical code. Numerical values are expressed in combinations of two-out-of-the-five bits (0, 1, 2, 4, 8) that are used to make up the code.

The following chart shows the numerical values and the corresponding two-out-of-five (2/5) code.

	0	1	2	4	8
0			X		X
1	X	X			
2	X		X		
3		X	X		
4	X			X	
5		X		X	
6			X	X	
7				X	X
8	X				X
9		X			X

Each position of the address register has five latches corresponding to the 0, 1, 2, 4 and 8 bits used in the 2/5 code. The IBM 1410 utilizes seven address registers.

The functions of these address registers are:

1. Instruction address register (IAR). This is used to locate and scan the instruction words.
2. A-address register (AAR). This locates the data word to be used as the A-field of an instruction.
3. B-address register (BAR). This locates the data word to be used as the B-field of an instruction. The A- and B-field addresses are read into the AAR and BAR respectively, when an instruction word is read out of storage.
4. C-address register (CAR). This usually contains the same address as the AAR at the beginning of the execute phase (E-phase).
5. D-address register (DAR). This usually contains the same address as the BAR at the beginning of the E-phase. The CAR and DAR are used in multiply, divide, recomplement, table search, and other operations.
6. E-address register, F-address register. These two registers contain the storage locations for characters going to or coming from I/O machines during overlapped operations. Overlapping is an optional feature and is covered elsewhere in this manual.

## Read-Out of Address Registers

The circuits for all address registers are very similar. For explanation purposes the 8-bit latch in the hundreds position of the BAR is used (Figure 2.2-1).

When the CPU requires the next character of the B-field, the RO BAR gates the output of every latch in the BAR, to the AR bus. All positions of the BAR read out at the same time. Set STAR sets the storage address register with the bits received on the AR bus. This causes the addressed character to be read out of storage.

## Read into Address Register

Before BAR is read into, the BAR reset line resets all the latches OFF. If the Addr Ch 8B line is up, the top leg of the AND circuits to all 8B latches is up. Set AR HP is up on the second leg of all hundred-position latches of all address registers. Set BAR is up to complete the coincidence necessary to set this latch.

The other line in Figure 2.2-1, STAR to IAR HP 8B, is only on the IAR latches and is used in instruction read-out.

### 2.2.02 Address Modification

Address modification is adding or subtracting one from the address in the STAR in order to develop the address of the next position of the word that is being read out of storage.

At the beginning of a storage cycle, the address of a position in storage is set into the STAR. During the storage cycle, this address is gated from the STAR through the address-modify unit one position at a time, starting with the units position (Figure 1.4-2). From the address modifier, the characters are gated on the address channel to an address register.

## Operation (Figure 2.2-2)

Address modification must be completed during the shortest storage cycle (4.5 microseconds). This storage cycle consists of six parts, called logic gates A through F. Each logic gate has a first and second clock pulse. During storage cycles longer than 4.5 microseconds, address modification is still completed from load gates A through F.

Consider a cycle where the A-address register (AAR) is used to read out a data word. The address in the AAR is set into STAR (Figure 2.2-3) on the second clock pulse of logic gate A. The AAR resets on the first clock pulse of logic gate B, while the units position of STAR is gated to the address modifier.

The address modifier controls are set to Modify By Minus One by First Scan Control. There are four scan-control latches in the CPU. The first and third scan latches are ON when scanning storage from units to the high-order positions. The second scan latch is ON when scanning from high- to low-order positions. No scan latch is ON when the address of the character is not to be changed.

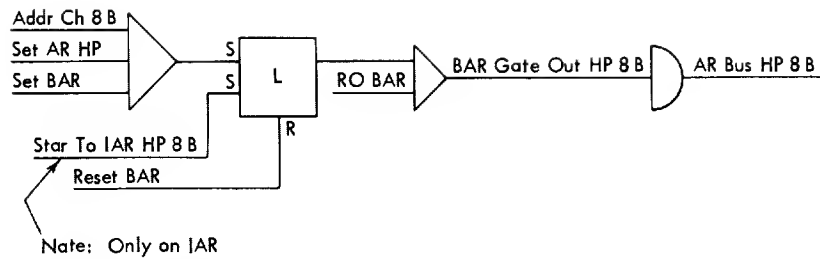


Figure 2.2-1 Address Register

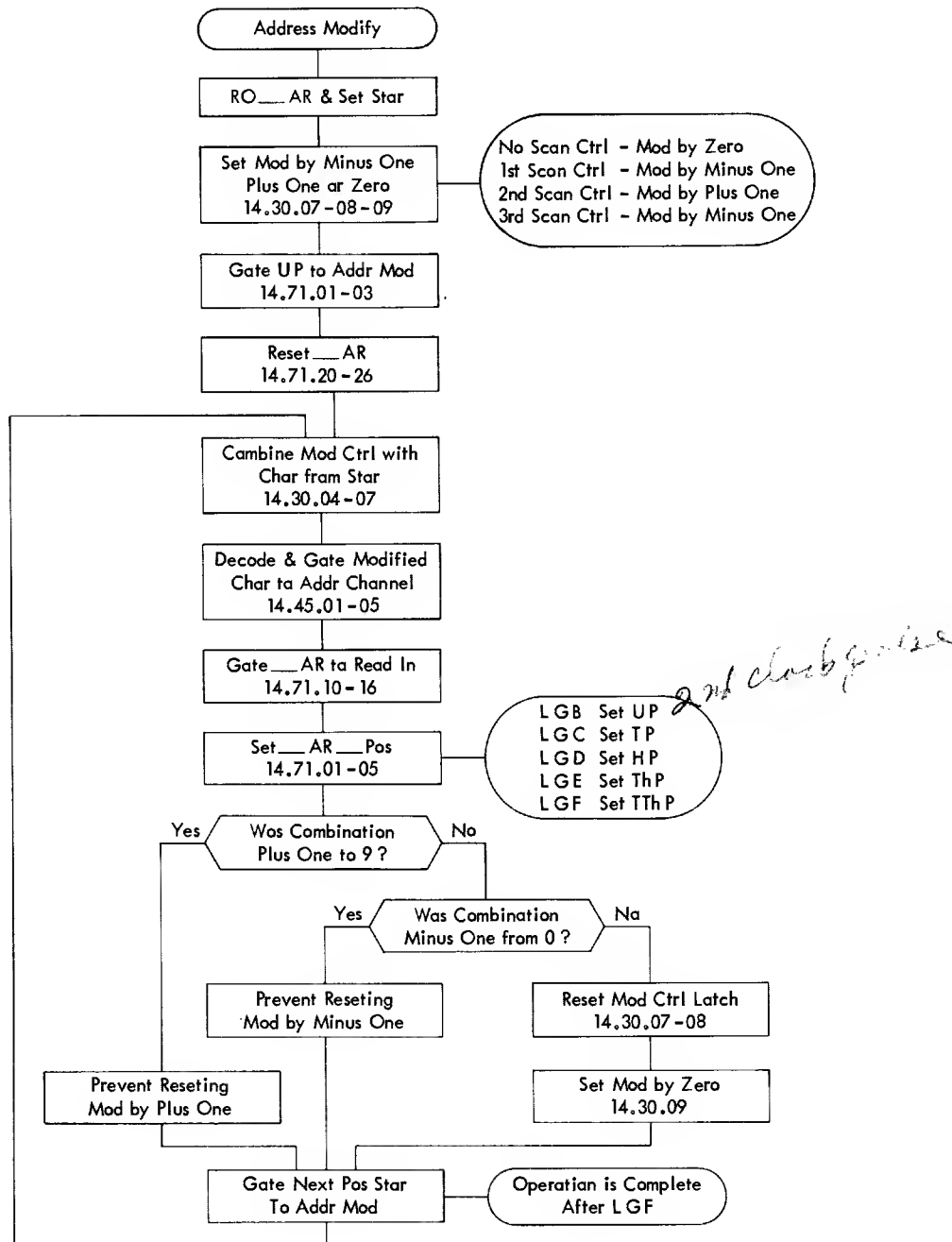


Figure 2.2-2 Address Modify



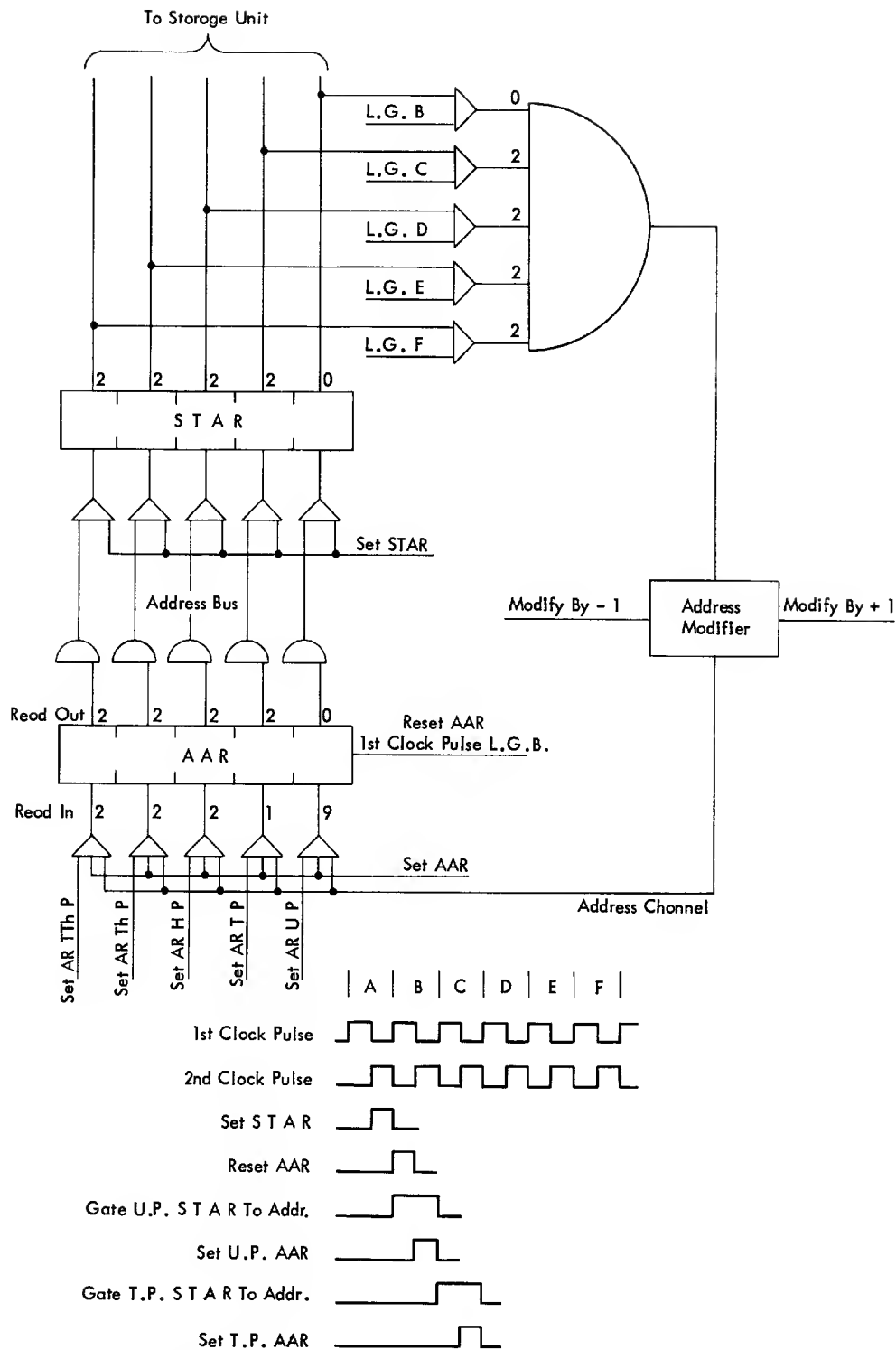


Figure 2.2-3 Address Modification

Modify by Minus One and the 0 from STAR (Figure 2.2-3) combine to bring up 8-bit and 1-bit lines that are gated into the units position of the AAR on the second clock pulse of logic gate B.

When the address modifier subtracts one from zero, it keeps the Mod by Minus One latch ON in order to borrow one from the next position. The 2 in the tens position of STAR is gated to the address modifier at logic gate C, and it combines with modify-by-minus-one to send a 1 to the tens position of the AAR.

The modifier controls are set to Modify by Zero. This passes the rest of the address through the modifier unaltered.

### 2.2.03 Address Exit-Channel Validity Check

The CPU utilizes two validity check (VC) circuits to validate the characters in the core-storage addresses. One VC circuit insures the validity of the characters when they are read in on the address channel. The other VC circuit checks the characters when they are gated from the address bus to the address exit channel.

#### Operation

Every valid character that is gated on the address exit channel falls into one of the two VC groups (Figure 2.2-4). If the character is invalid because a bit was dropped neither of the VC groups are conditioned. If the character has an extra bit both VC groups are conditioned at the same time. The VC groups gate on the error trigger when they are either both down or up. If the trigger is ON at the end of the cycle (error sample time) Addr Exit Error is gated to stop the CPU with this error (master error).

The input to the VC circuit must be a valid character on every storage cycle. When the exit channel is not being used, zeros are inserted on the channel to satisfy the circuit.

The check test lines in Figure 2.2-4 are controlled by three momentary switches on the console CE panel. These switches, labeled first, second, and third check test, are used to force error conditions to insure that the transistors in the check circuits are working properly.

Check the circuits by pressing a check test switch and the start key. The CPU takes a cycle and should stop with every check-indicator light ON.

The first check test conditions both VC groups to cause an error. The second check test blocks the zero inserted on the exit channel to cause both VC groups to be down. The third check test gates the error trigger ON. Every transistor in the check circuit must operate correctly or the error light will not come on.

### 2.2.04 Address-Channel Validity Check

The circuit for this VC is similar to the exit-channel circuit except that a new character is gated on the channel every logic gate instead of one character each cycle. An error must be stored until the end of the storage cycle.

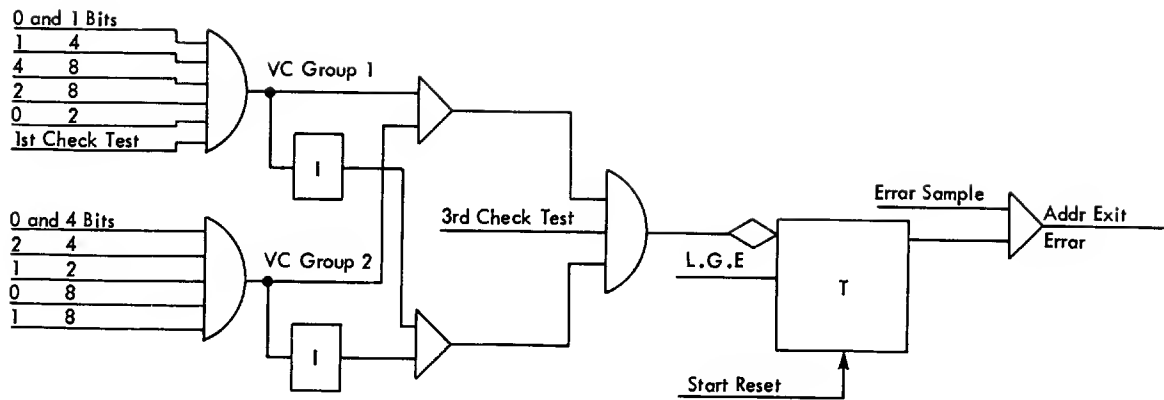


Figure 2.2-4 Address Exit Validity Check

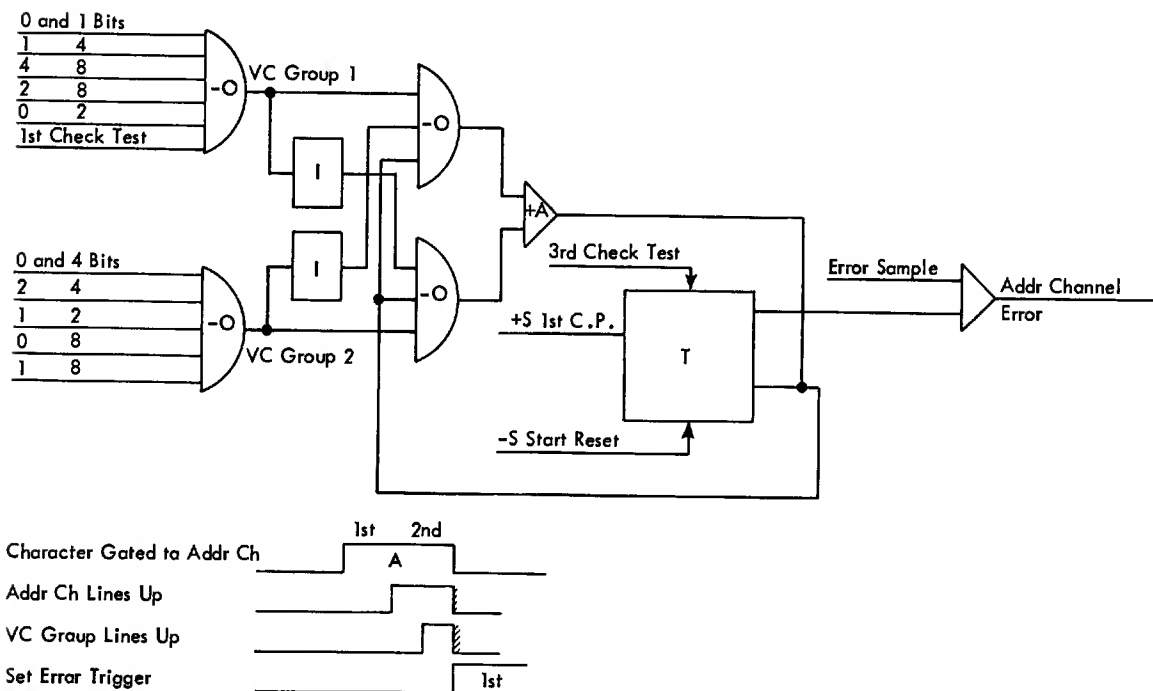


Figure 2.2-5 Address Channel Validity Check

## Operation

Each character falls into one of two VC groups. If both or neither group is up, an error must be detected. Because there is a new character on each logic gate, the circuit must allow time for the character to arrive on the channel and to be decoded into one of the two groups before the VC group lines can be sampled. The circuit allows time by controlling a trigger with collector pullover.

The AND circuit (Figure 2.2-5) shares a common load resistor with the OFF side of the error trigger. Before the trigger can be set ON the transistor in the AND circuit must be cut off. Once the trigger is set ON, the OFF side goes minus to satisfy the NOR circuits and to keep the AND circuit transistor cut off. Regardless of the validity of the remaining characters that are gated onto the address channel, the trigger remains ON, and at the end of the cycle Address Channel Error is gated to stop the CPU.

On storage cycles longer than 4.5 microseconds, a zero is inserted on the address channel to satisfy the VC circuit. The second check test blocks the zero to force both VC groups down in order to cause a VC error.

### 2.3.00 CONTROL UNIT

#### 2.3.01 Cycle Control

Storage cycles are identified by cycle latches that are labeled A- through F-, I-, or X-cycle. Type of cycle and operation determine the length of the cycle. The cycle latch is on from logic gate B to logic gate B. Cycle latches are used to control different areas of the CPU. For example, the I-cycle latch identifies the character that is being read out of storage as part of an instruction word. I-cycle also gates the modified address into the IAR.

Each cycle latch has a control latch that is usually set at the next-to-last logic gate of the previous cycle. For example, at the end of the I-cycle at I-ring Optime the I-Cycle Ctrl latch is set to keep the I-cycle latch ON for the next storage cycle.

The A-cycle latch usually identifies the character that is being read out of storage as an A-field character. The B-cycle usually identifies the character as a B-field character. E- and F-cycles are used for input-output operations. C-, D-, and X-cycles are used for special operations.

There are four more control latches that assist in identifying characters as they are read out of storage. They are called units, body, extension, and MQ.

For example, consider an add operation where the contents of the A-field are added, a digit at a time, units position first, to the contents of the B-field. The first cycle after I-phase is an A-cycle during which the units position of the A-field is set into the A-data register. The next cycle is a B-cycle during which the units position of the B-field is read out onto the B-channel. The units latch is set for the first A- and B-cycles to identify the characters as units-position characters because the signs of the fields are located there.

The body latch is set after the units position is set, and stays ON until the end of the A-field. The extension latch identifies the remaining B-field characters that have no corresponding A-field characters.

	E	B	U
	X	O	N
	T.	D	I
		Y	T
			S
A-Field		0 1 2 9	<sup>+</sup> 0
B-Field	0 2	1 9 8 7	<sup>+</sup> 0
	0 2	2 1 1 6	<sup>+</sup> 0

The MQ latch is used in multiplication and division to identify a character as a position of the multiplier or quotient. The MQ latch is also used in edit and zero suppress Op codes.

### 2.3.02 Logic Clock

The logic clock uses a crystal-controlled oscillator and a ten-trigger ring to develop the number of logic gate pulses required for each storage cycle.

The oscillator drives a binary input trigger called the clock pulse trigger. The outputs of the clock pulse trigger are called first clock pulse and second clock pulse. The clock is normally stopped with the stop latch that, by collector pull-over, holds the clock pulse trigger at first clock pulse. Logic gate A-trigger is also ON. When the stop latch is reset, the clock pulse trigger is advanced by the oscillator. At the end of second clock pulse (2 cp), the ring is advanced from logic gate A (LGA) to logic gate B (LGB). (See Figure 2.3-1.) Since the clock pulses are .375 microseconds long, the ring advances every .75 microseconds.

Type of cycle and of operation control the length of the cycle by conditioning the stop at F, G, H, J, or K-line. This line combines with the corresponding logic gate pulse to bring up Last Logic Gate. Last Logic Gate sets the logic gate F-latch that gates the ring back to LGA. To stop the clock at the end of a cycle, the stop latch is set on the 2 cp of the last logic gate (LLG). The stop latch combines with 1 cp to hold the clock pulse trigger at 1 cp, LGA. All I/O commands (except F or K) can be executed in either of two fashions: unoverlapped or overlapped. An unoverlapped command causes the CPU to take E- or F-cycles, as they are required, to read out or to store each I/O character. Because the CPU can store or read out characters at a faster rate than any I/O machine, the CPU must wait between each E- or F-cycle.

During E-phase of an unoverlapped I/O instruction, Disable Compute Cycle comes up to stop the clock by preventing (by collector pullover) the LGA trigger from coming on. The clock pulse trigger continues to advance. When an I/O cycle is needed, E- or F-Cycle Required blocks Disable Compute Cycle to allow the ring to advance through the cycle. Logic gate Z that gates LGA ON, is reset by LGA. As long as Disable Compute Cycle prevents LGA from setting, LGZ remains ON (Figure 2.3-2).

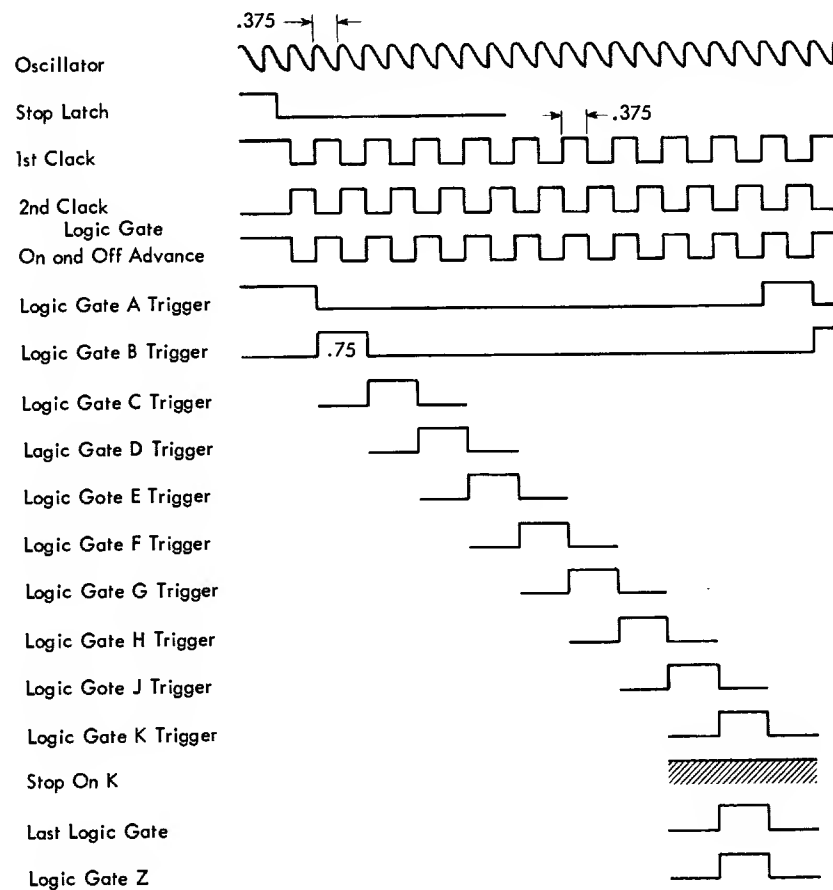
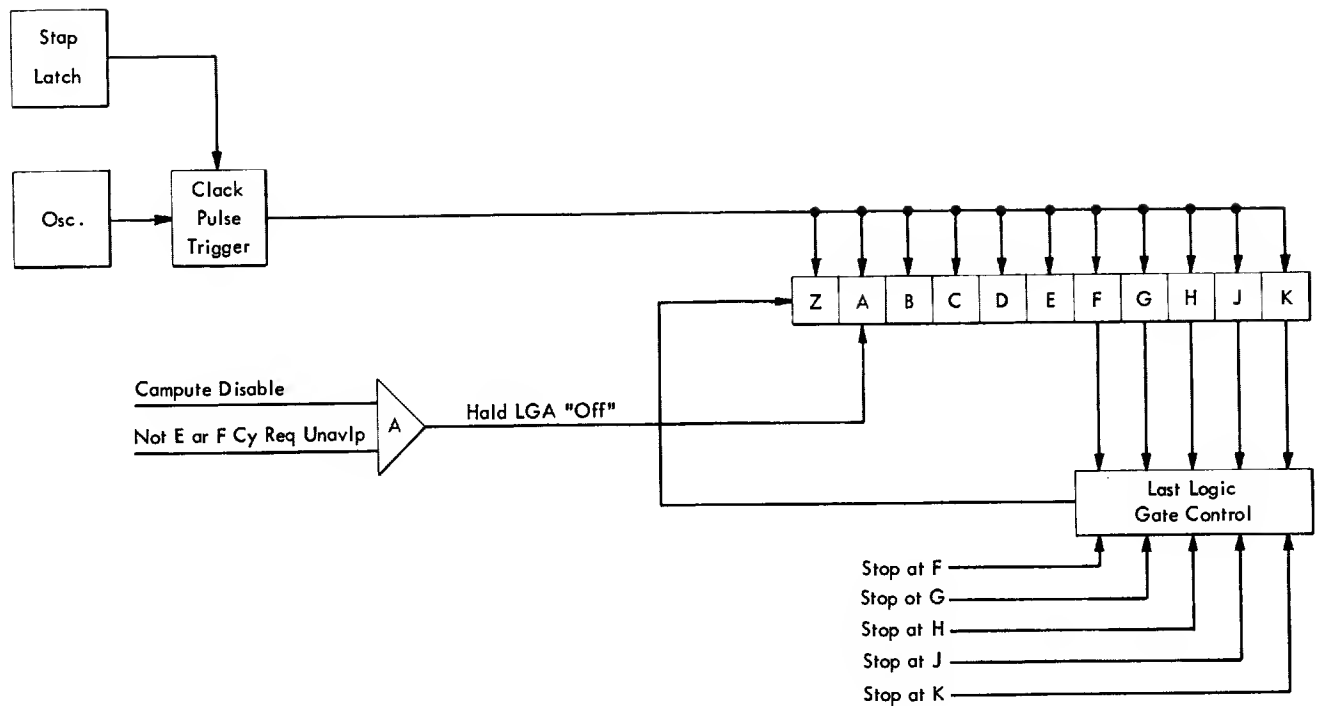


Figure 2.3-1 Logic Clock



When an I/O instruction overlaps, it allows the CPU to use the time between E- or F-cycles for compute cycles.

Eight triggers are added to the clock circuits to form a logic gate extension ring. These triggers are called LGR through LGY. When an overlapped cycle is taken, the extension ring is used instead of the normal logic-clock ring.

Logic gate Z gates ON both LGA and LGR. When E-Cycle Required or F-Cycle Required is up in overlapped operation, it conditions the Disable Compute Cycle to hold LGA OFF, while LGR is allowed to set (Figure 2. 3-3). When the I/O cycles are not needed, LGR is held OFF and LGA is allowed to set.

## 2. 4. 00 OPERATIONAL UNIT

### 2. 4. 01 A-Data Register

The A-data register is a single-character storage register capable of storing any valid BCD coded character with or without word marks.

Characters are read into the A-data register from storage via the B-channel, or from an address register via the AR Exit Channel (Figure 2. 4-1). The A-data register is gated out onto the A-channel to the adder, compare, and assembly units. The A-data register is reset (all OFF, except the C-bit to maintain proper parity) before it can be read in.

#### A-Character-Set Check

The A-register-set check insures that the A-data register receives a reset impulse whenever it is read in.

Two binary input triggers (18. 14. 07) must both be OFF or ON at the end of every cycle, or an error is indicated. One trigger is impulsed whenever Sw B Ch to A-reg or Sw AR Exit Ch to A-reg are up. The other trigger must receive the Reset A-Data Reg impulse before the end of the cycle or the error stops the machine.

### 2. 4. 02 B-Channel Validity Check

Every character read out of storage is available on the B-channel. The B-channel validity-check (Figure 2. 4-2) circuit insures that each character has an odd number of bits. The numerical bits are combined to bring up numerics odd or numerics even line. The zones, WM, and C-bits are combined to bring up odd of even. Numerics odd or numerics even combines with zones WM C-odd or WM C-even to bring up character-odd. An error is indicated if the character is either even or not odd.

The errors detected by character-even are:

1. Bit configuration even.
2. Any bit with bit and not bit lines both up or both down.
3. Any transistor failure in the check circuit which brings a character-even line up.



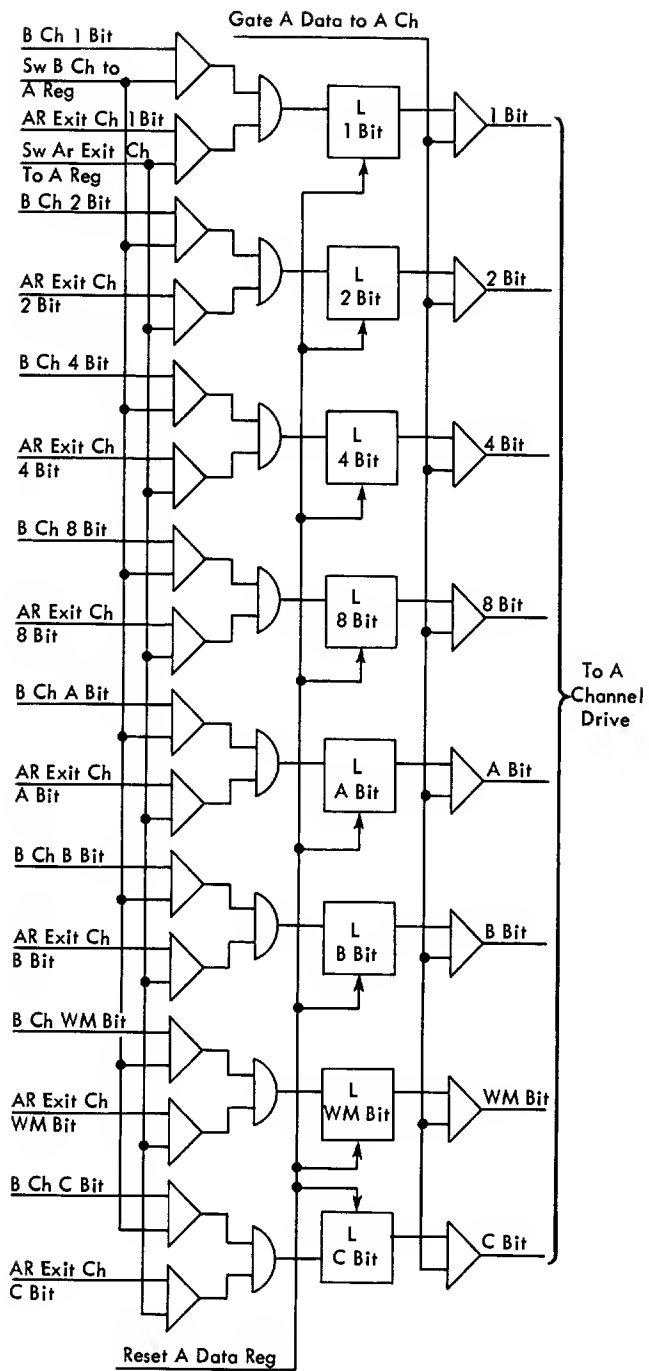


Figure 2.4-1 A Data Register

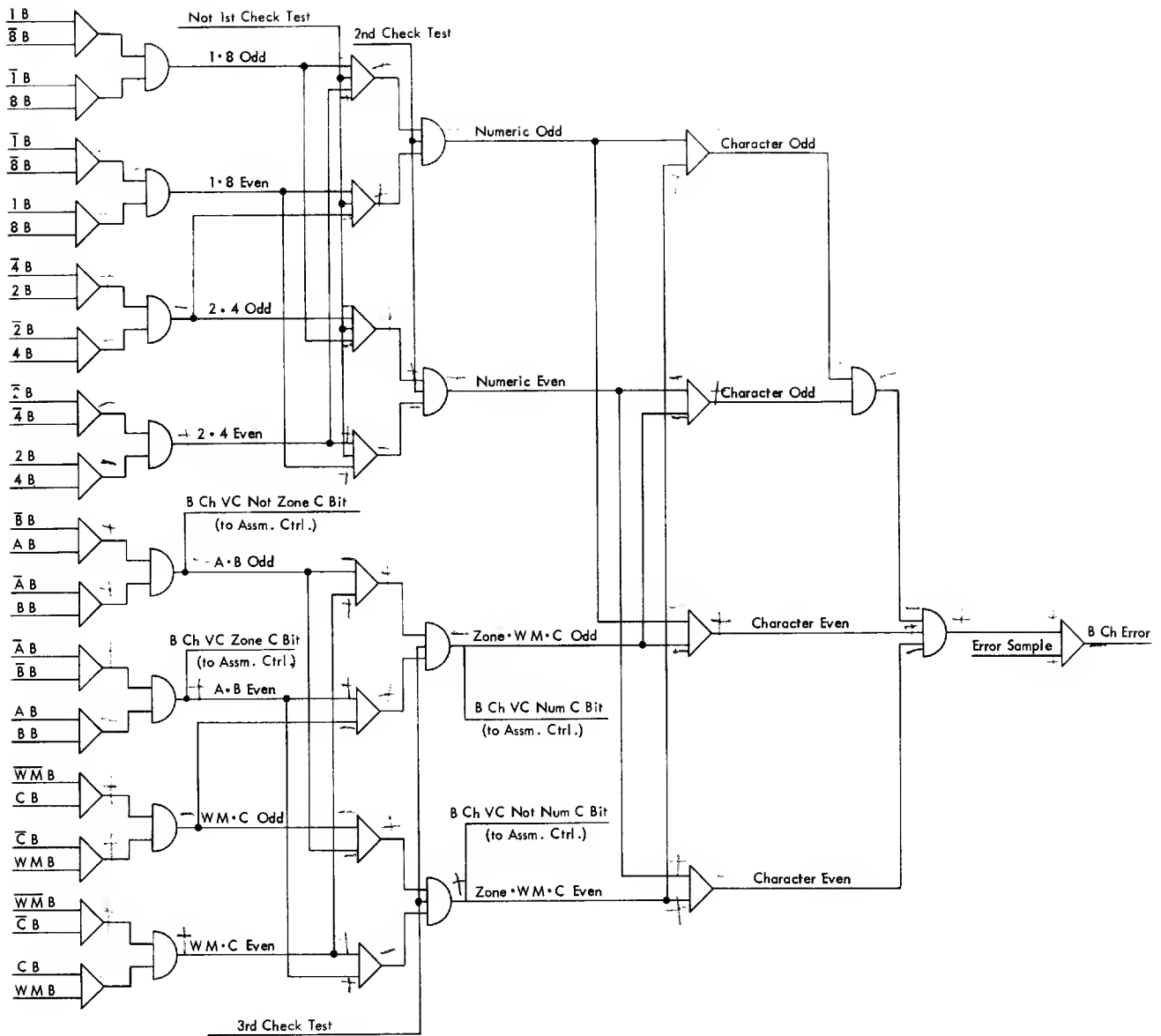


Figure 2.4-2 B-Channel Validity Check

The errors detected by character-not-odd are:

1. Any transistor failure that prevents the digits-odd or digits-even lines from coming up.
2. Any transistor failure that prevents the zone, WM, C-bit-even, or C-bit-odd lines from coming up.

Any of the last seven transistors (Figure 2.4-2) could fail without being detected. These transistor failures are detected by three check tests.

1. 1st Check Test blocks both numerics odd and numerics even. Without numerics odd or even neither character-odd nor character-even can come up.
2. 2nd Check Test conditions both numerics odd and numerics even. These combine with zones WM C-odd or WM C-even (depending on the character on the channel) to bring up both character-odd and character-even.
3. The third check test conditions both zones WM C-odd and WM C-even to combine with numerics odd or numerics even to again bring up both character-odd and character-even.

#### 2.4.03 A-Channel

Information is gated to the A-channel from one of four registers: A-, E- or F-data registers, or Op Mod Register. The A-channel feeds data to the A-side of the adder, compare and assembly units.

##### A-Character-Select Check

The A-character-select check (18.14.01) insures that only one of the four registers at a time is gated to the A-channel.

##### A-Channel Validity Check

The A-channel validity check circuit is similar in operation to the B-channel validity check. The logic circuits are different (Figure 2.4-3).

#### 2.4.04 Assembly

The assembly unit receives data from five areas of the CPU and combines the data as determined by the operation. The five inputs to the assembly are the A- and B-channels, the adder, the special-character generator, and the zone adder used for 1401 compatibility.

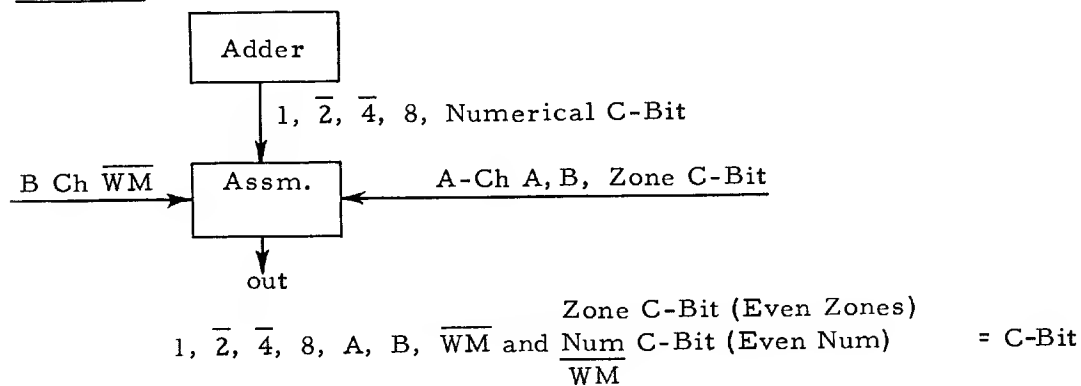
In the assembly, each character has three parts: the numerical part (1, 2, 4, and 8-bits), the zones (A and B-bits), and the word mark. To gate a character through the assembly requires three controls: numerical, zones, and word mark. For example: assume an input operation where the data enters the CPU as an 80-character record. The storage locations into which the record is to be placed, contain word marks that define the fields of the record. The assembly is controlled to Use A Ch Num, Use A Ch Zones, and Use B Ch WM. When a word mark from storage is placed on the B-channel, the assembly unit combines it with the

input character on the A-channel. The A-channel character with a word mark is put on the assembly channel where it is read into storage.

During an arithmetic operation, a character can be the combination of the adder (Use Adder Nu), the sign stored in the sign latches (Use Sign Latches), and a word mark from the B-channel.

To maintain the proper parity of the new character, the assembly must generate a C-bit depending on the number of bits in the three parts of the new character. To assist in determining whether a C-bit is required, the numerical and zone portions, at the inputs to the assembly, contain other bits called the numerical C-bit and zone C-bit. These C-bits retain odd parity for their respective portions of the character. For example, the adder output generates a numerical C-bit along with the 1- and 8-bits for a nine. The numerical and zone C-bits for the A- and B-channels are developed in their respective validity-check circuits (Figures 2.4-2 and 2.4-3). As assembly controls select the portions of characters to be gated through the assembly, they also select the numerical and zone C-bits that are to be combined with the WM-bit to develop an assembly-channel C-bit.

Example:



Other assembly controls are:

Use B Ch WM  
Use B Ch Zones  
Use B Ch Nu  
Use A Ch WM  
Use A Ch Zones  
Use A Ch Nu  
Use No WM  
Use No Zones  
Use No Nu  
Use Adder Nu  
Use Sign Latch  
Use A Ch Sign  
Use B Ch Sign  
Invert A Ch Sign  
Invert B Ch Sign

These generate the sign in the standard form, that is, a  $\overline{BA}$ -bit for minus and AB for plus.

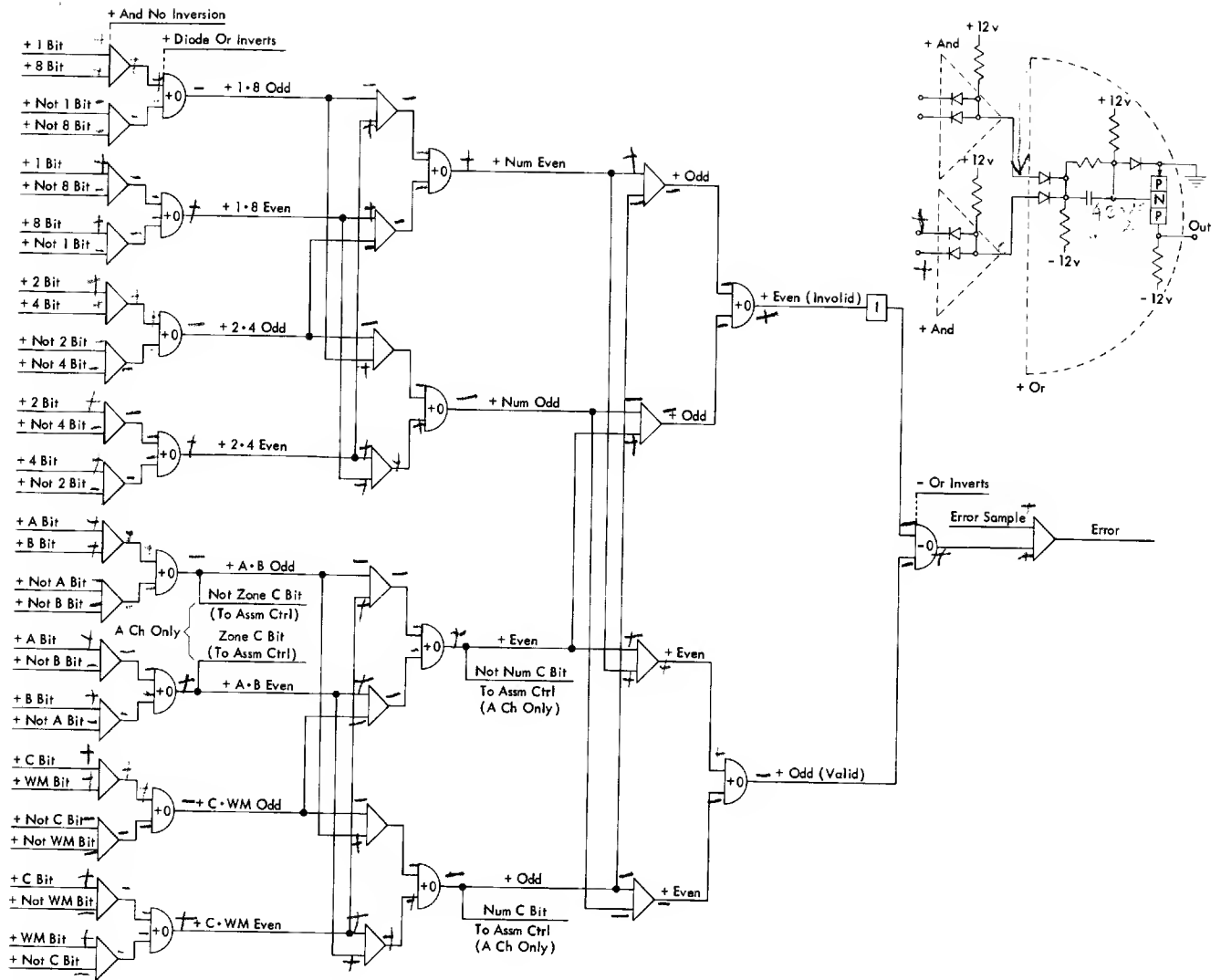


Figure 2.4-3 A-Channel Validity Check

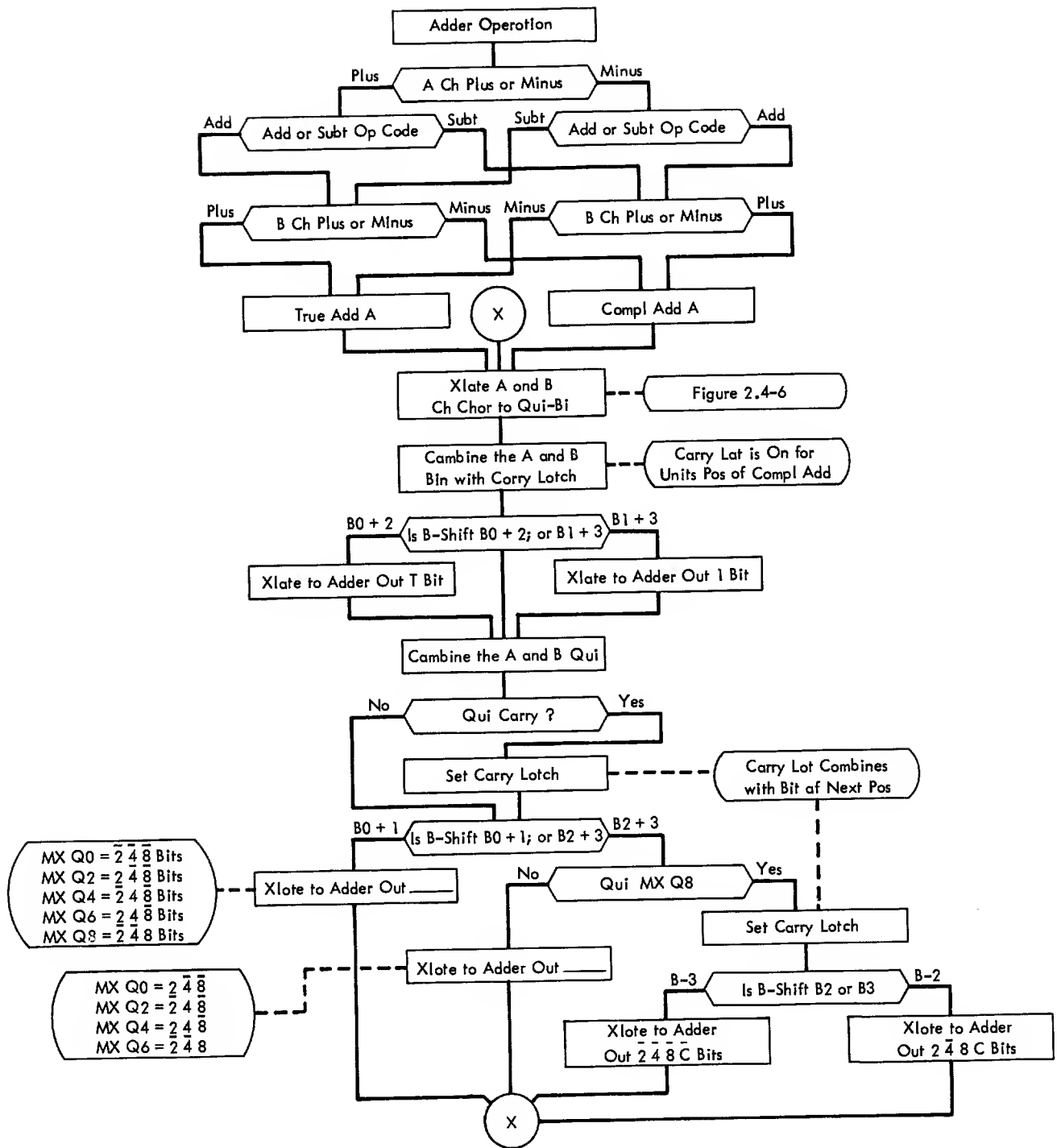


Figure 2.4-4 Adder Operation

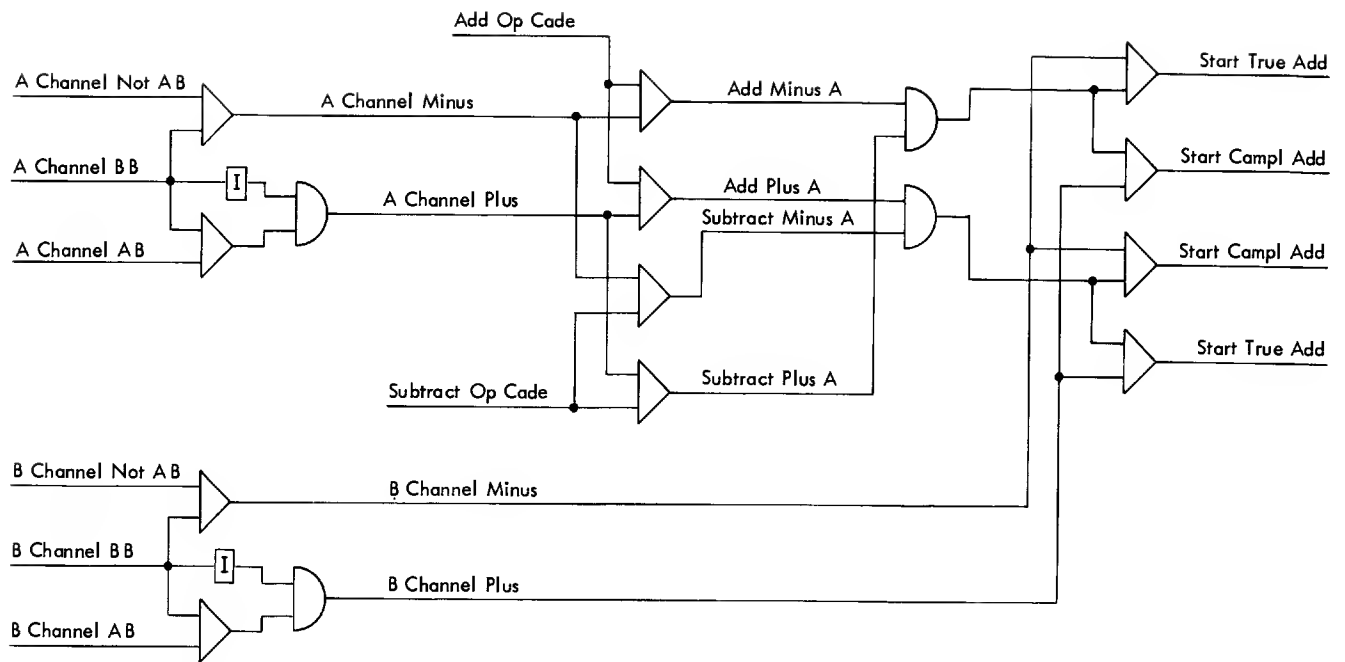


Figure 2.4-5 True Complement Controls

Card Code	BCD	True Add	Complement Add
0	8-2	B0 Q0	B1 Q8
1	1	B1 Q0	B0 Q8
2	2	B0 Q2	B1 Q6
3	1,2	B1 Q2	B0 Q6
4	4	B0 Q4	B1 Q4
5	1,4	B1 Q4	B0 Q4
6	2,4	B0 Q6	B1 Q2
7	1,2,4	B1 Q6	B0 Q2
8	8	B0 Q8	B1 Q0
9	1,8	B1 Q8	B0 Q0

Figure 2.4-6 Qui-Binary Code

Set WM	
Use <u>Assm 0 Insert</u> (This generates 8, 2, and numerical C-bits.)	
Generate <u>Num One</u> (This generates 1 not-numerical C-bit.)	
Generate <u>A-bit</u> (This generates A not-zone C-bit.)	
Generate <u>Dollar Sign</u>	} These require no other assembly controls.
Generate <u>Asterisk</u>	
Generate <u>Group Mark WM</u>	

#### Assembly Channel Validity Check

This check circuit is the same as the A-channel validity-check circuit (Figure 2. 4-3).

#### 2. 4. 05 Adder

The IBM 1411 Central Processing Unit uses a single-digit adder to perform all arithmetic operations. The adder receives digits from the A- and B-channels and combines them algebraically. The results of the addition are routed through the assembly area back to the storage unit.

#### Operation (Figure 2. 4-4)

When two fields are added together, they are fed to the adder units position first, because that is the position that contains the sign of the field.

The sign is indicated by the zone bits over the units position. The four possible combinations of these zones are:

12 zone	A- and B-bits	positive sign
11 zone	B-bit only	negative sign
0 zone	A-bit only	positive sign
No zones	Neither A- nor B-bits	positive sign

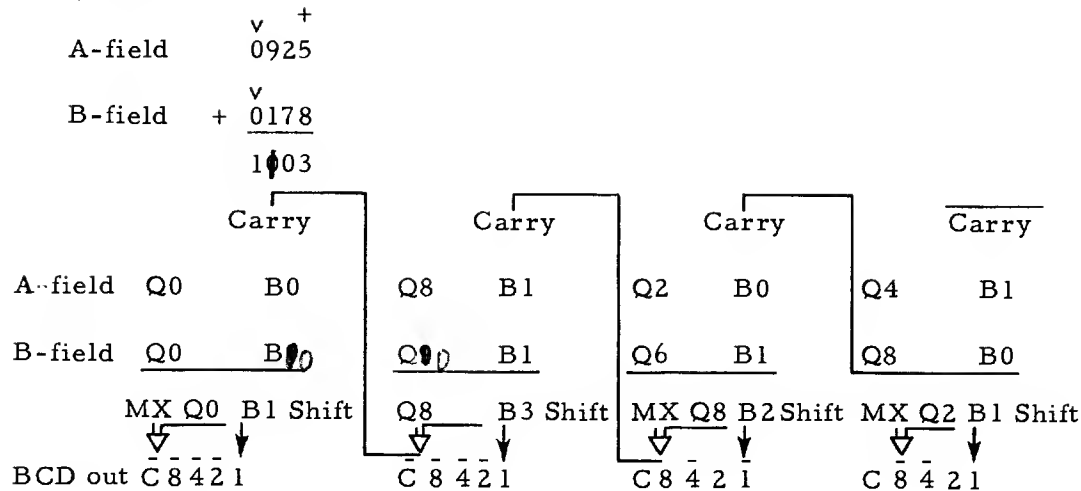
When both the A- and B-channels contain the units-position characters, the signs of the fields are analyzed along with the type of operation in order to determine whether a true or complement (T/C) add is required (Figure 2. 4-5).

The digits from the A- and B-channels are translated from the BCD code to the qui-binary code in either true or complement form as determined by the T/C add controls (Figure 2. 4-6).

The adder matrix combines the binary portion of the digits along with a carry, if required. The quinary portions of the digits are also combined. The result is translated, along with the binary output, to the BCD code in the adder-output translator.



Example:



## 2.4.06 Compare

The compare unit receives information from the A- and B-channels. The two characters are compared to determine whether the B-channel character is greater than ( $>$ ), equal to ( $=$ ), or smaller than ( $<$ ) the A-channel character. Figure 2.4-7 shows the characters and codes in the collating sequence from low to high character. The result of the compare sets either a high, low, or equal latch that can be tested by later program instruction.

### Operation

The two characters are fed to both the compare and adder units. In the compare unit, the characters are translated into one of three groups: alphameric (AN), special character (SC), or non-numerical (NN) (Figure 2.4-7). Also in the compare unit, the zones are compared. In the adder, the digits are combined to determine whether the B-channel digit is greater than, equal to, or smaller than the A-channel digit.

The compare unit first analyzes the groups that the two characters fall into. For example, if one character is AN and the other is either NN or SC, the AN character is high regardless of the zones or digits involved.

If the groups alone do not reveal the relationship of the two characters, the comparison of the zones is analyzed.

If the relationship of the character is still unknown, the result of the adder comparison must be considered.

The adder compares the two digits by subtracting the A-channel digit from the B-channel digit.

A carry or no-carry is developed when the quinary portions of the A- and B-channel characters are combined, except when the result is a quinary 8. If a carry is called for, it signals that the B-channel character is larger than the A-character.

	Character	Card Code	BCD Code						
			C	B	A	8	4	2	1
NN		No Punches	X						
	.	12-3-8		X	X	X		X	X
SC	□	12-4-8	X	X	X	X	X		
	(	12-5-8		X	X	X	X		X
	<	12-6-8		X	X	X	X	X	
	≠ (G/M)	12-7-8	X	X	X	X	X	X	X
NN	&	12	X	X	X				
	\$	11-3-8	X	X		X		X	X
	*	11-4-8		X		X	X		
SC	)	11-5-8	X	X		X	X		X
	;	11-6-8	X	X		X	X	X	
	△	11-7-8		X		X	X	X	X
NN	-	11		X					
	/	0-1	X		X				X
	,	0-3-8	X		X	X		X	X
	%	0-4-8			X	X	X		
SC	= (W/S)	0-5-8	X		X	X	X		X
	,	0-6-8	X		X	X	X	X	
	"	0-7-8			X	X	X	X	X
NN	¢	2-8			X				
	#	3-8				X		X	X
	@	4-8	X			X	X		
SC	:	5-8				X	X		X
	>	6-8				X	X	X	
	√ (T/M)	7-8	X			X	X	X	X
	?	12-0	X	X	X	X		X	
	A	12-1		X	X				X
	B	12-2		X	X			X	
	C	12-3	X	X	X			X	X
	D	12-4		X	X		X		
	E	12-5	X	X	X		X		X
	F	12-6	X	X	X		X	X	
	G	12-7		X	X		X	X	X
	H	12-8		X	X	X			
	I	12-9	X	X	X	X			X
	!	11-0		X		X		X	
	J	11-1	X	X					X
	K	11-2	X	X				X	
AN	L	11-3		X				X	X
	M	11-4	X	X			X		
	N	11-5		X			X		X
	O	11-6		X			X	X	
	P	11-7	X	X			X	X	X
	Q	11-8	X	X		X			
	R	11-9		X		X			X
	≠ (R/M)	0-2-8			X	X		X	
	S	0-2	X		X			X	
	T	0-3			X			X	X
	U	0-4	X		X		X		
	V	0-5			X		X		X
	W	0-6			X		X	X	
	X	0-7	X		X		X	X	X
	Y	0-8	X		X	X			
AN	Z	0-9			X	X			X
	0	0	X			X		X	
	1	1							X
	2	2						X	
	3	3	X					X	X
	4	4					X		
	5	5	X				X		X
	6	6	X				X	X	
	7	7					X	X	X
	8	8				X			
	9	9	X			X			X

Figure 2.4-7 Collating Sequence

Example: (B-character, 5. A-character, 3.)

B	Q4	B1	
A	Q6	B0	(9's complement)
		carry	(1st cycle of CA forces a carry)
<hr/>		<hr/>	
carry	Q0	B2	
↑			
signifies B > A			

If no-carry is brought up, the B-channel character is smaller than the A-character. Example: (B-character, 3. A-character, 7.)

B =	Q2	B1	
A =	Q2	B0	
		carry	
<hr/>		<hr/>	
no-carry	Q4	B2	
signifies B < A			

When the quinary result is a Q8, the combined binary portion of the character determines the result of the compare. Example: (B-character, 6. A-character, 6.)

B =	Q6	B0	
A =	Q2	B1	
		carry	
<hr/>		<hr/>	
	Q8	B2	signifies A = B

Example: (B-character, 8. A-character, 9.)

B =	Q8	B0	
A =	Q0	B0	
		carry	
<hr/>		<hr/>	
	Q8	B1	signifies B < A

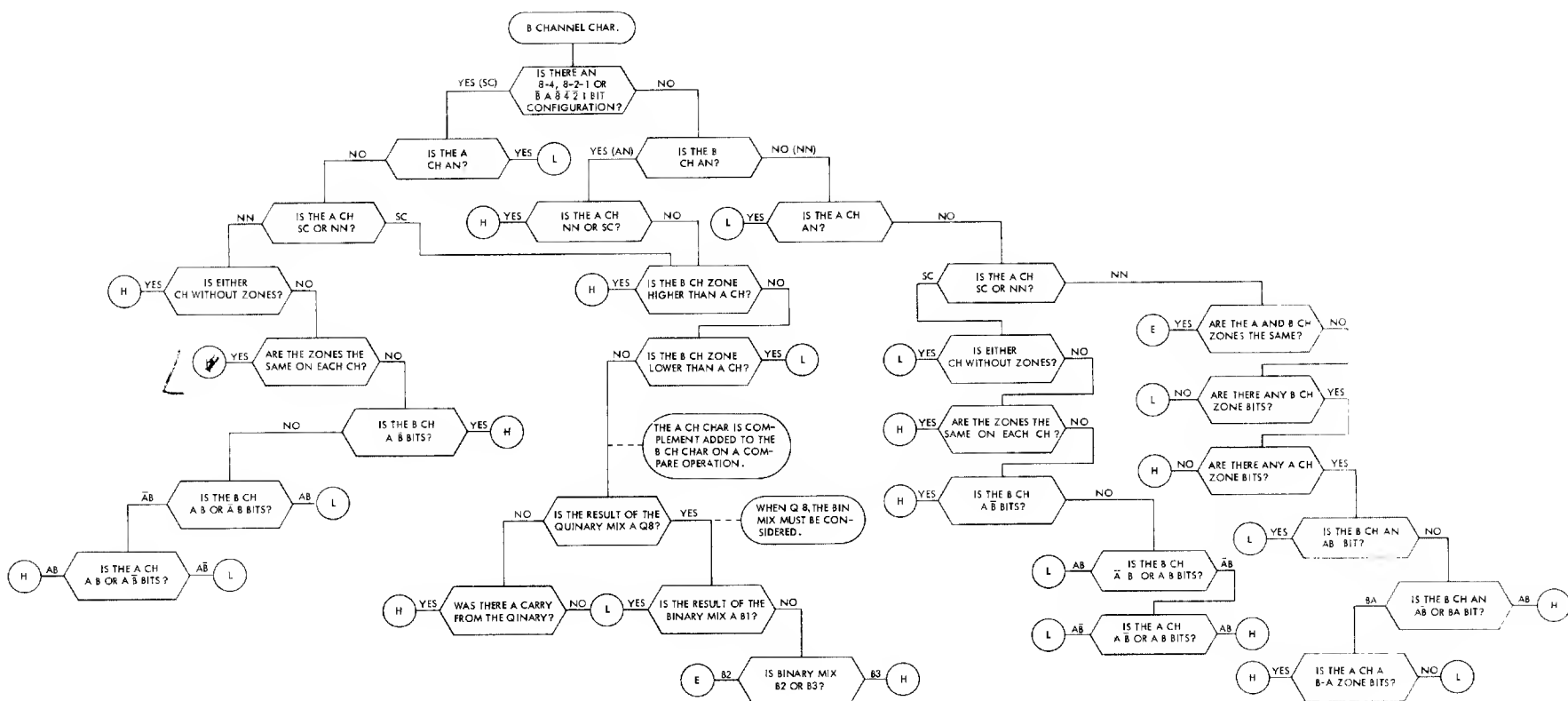
Example: (B-character, 5. A-character, 4.)

B =	Q4	B1	
A =	Q4	B1	
		carry	
<hr/>		<hr/>	
	Q8	B3	signifies B > A

Special characters 8-3 through 8-7 are added as a 3 through 7 respectively.

To determine whether the compare unit operates correctly, select two characters and follow them through the flow chart (Figure 2.4-8). The translation of the A-field character into the three groups is the same as shown for the B-channel character.

Figure 2.4-8 Compare Unit Operation



## 2.5.00 RESETS

### Start Reset

The start-reset pulse resets the master controls to allow the operator to proceed through a program, if a master error stops the operation. Start reset is initiated by the start-key pulse. The start-key pulse is only available when the start key is pressed when in the run mode, I/E mode, or storage scan mode (after the address set routine).

The various latches and triggers that are reset by the start reset are:

<u>Control</u>	<u>Status</u>	<u>Logic</u>
Address check latch	Off	18.14.11
Error sample latch	Off	18.14.08
Address ch error trigger	Off	18.14.03
Address exit error trigger	Off	18.14.02

### Program Reset

Pressing the program reset key causes the following operations to take place.

1. The cycle control, scan control, and scan latches are reset.
2. The I- and A-ring triggers are reset.
3. The Op register, Op Mod Register, and the A-data register are reset to contain C-Bits.
4. The branch to 00001 latch is set.
5. The storage address register is reset.
6. The set check circuits are reset.
7. The 1401 control latches are reset.
8. The arithmetic control latches are reset.
9. The logic clocks are reset.
10. Program reset causes a start reset.

The program reset is developed as follows:

<u>Signal</u>	<u>Control</u>	<u>Logic</u>
Program Reset	<u>Prog Rst Key</u> (Logic clock) stopped at A	12.65.04

Following is a listing of the various latches and triggers that are reset by the program reset.

Program Reset 1 (12.65.04)

<u>Control</u>	<u>Status</u>	<u>Logic</u>
X-cycle latch	Off	12. 12. 05
A-cycle ctrl latch	Off	12. 12. 20
C-cycle ctrl latch	Off	12. 12. 20
B-cycle ctrl latch	Off	12. 21. 21
D-cycle ctrl latch	Off	12. 12. 21
I-cycle ctrl latch	On	12. 12. 23
X-cycle ctrl latch	Off	12. 12. 23
No-scan latch	Off	12. 30. 01
1st-scan latch	Off	12. 30. 01
2nd-scan latch	Off	12. 30. 02
Spl adv ctrl latch	Off	12. 13. 01
I-ring ctrl latch	On	12. 13. 01
No-scan ctrl latch	Off	12. 30. 03
1st-scan ctrl latch	Off	12. 30. 03
2nd-scan ctrl latch	Off	12. 30. 04
3rd-scan ctrl latch	Off	12. 30. 04
I-ring triggers	Off	11. 20. 01-07
A-ring triggers	Off	14. 70. 01-06
Console WM input latch	Off	15. 60. 09
* Op reg C-bit latch	On	13. 10. 01
* Op reg A, B, 8, 4, 2, and 1 bit latches	Off	13. 10. 01
Binary reg bit latches	Off	14. 18. 01-02
Memory addr reg latches	Off	14. 17. 01-15
Index tag latches	Off	14. 50. 01-02
A-data reg latches (A, B, 8, 4, 2, 1 bits and WM)	Off	15. 39. 01-08
A-data reg C-bit latch	On	15. 39. 07
Op mod C-bit latch	On	13. 12. 01
Op mod A, B, 8, 4, 2, and 1 bit latches	Off	13. 12. 01-03

\* The Op register is reset by gating the Set Op Reg Pulse. No character is gated to the B-channel because neither Memory Load nor Regen is conditioned. Because no bits are on the B-channel, the Set Op Reg Pulse resets all of the Op register latches except the C-bit latch that is set ON by inverting the Not B-Ch C-Bit line.

Program Reset 2 (12. 65. 06)

<u>Control</u>	<u>Status</u>	<u>Logic</u>
I/O percent latch	Off	13. 50. 01
I/O coml AT latch	Off	13. 50. 01
I/O lozenge latch	Off	13. 50. 02
1401 print error latch	Off	13. 65. 01
1401 file validity-check latch	Off	13. 65. 02
1401 wrong-length record latch	Off	13. 65. 02
1402 file address compare latch	Off	13. 65. 02
1401 end-of-reel latch	Off	13. 65. 03
1401 tape error latch	Off	13. 65. 03
1401 process check latch	Off	13. 65. 03
1401 inquiry error latch	Off	13. 65. 08
1401 punch trigger	Off	13. 70. 02
1401 I/O end	Off	13. 70. 02
1401 print trigger	Off	13. 70. 01
1401 read trigger	Off	13. 70. 01
1st I/O cycle control trigger	Off	13. 70. 03
E-cycle latch	Off	12. 12. 66
F-cycle latch	Off	12. 12. 67

Program Reset 3 (12. 65. 05)

<u>Control</u>	<u>Status</u>	<u>Logic</u>
Logic gate triggers (A-U)	Off	11. 10. 11-23
Console set start cnd latch	Off	12. 15. 03
Process routine latch	Off	12. 15. 03
No-branch latch	Off	12. 60. 14
Branch to A-AR latch	Off	12. 60. 14
Branch to 00001 addr latch	On	12. 60. 14
Set op reg trigger	Off	18. 14. 04
Check op reg trigger	Off	18. 14. 04
Reset mem data reg trigger	Off	18. 14. 06
1st trigger check	Off	18. 14. 06
Set A-data reg trigger	Off	18. 14. 07
Reset A-data reg trigger	Off	18. 14. 07

Program Reset 4 (12.65.05)

<u>Control</u>	<u>Status</u>	<u>Logic</u>
Console reset-start condition latch	Off	41. 10. 01
Address set-routine latch	Off	41. 10. 02
Display routine latch	Off	41. 10. 03
Storage scan latch	Off	41. 10. 03
WM cnd alter latch	Off	41. 20. 01
Full-line cnd alter latch	Off	41. 20. 01
Console stop condition latch	Off	44. 10. 01
Console stop print latch	Off	44. 10. 02
Console clock trigger	Off	45. 10. 01
Console cycle matrix triggers	Off	45. 20. 01
Console cycle latch	Off	45. 50. 01
Cnd res console printer not busy latch	On	45. 50. 01
Console check strobe trigger	Off	45. 50. 01
Console printer strobe trigger	Off	45. 50. 02
Reset console printer not busy latch	Off	45. 50. 03
Solenoid driver strobe latch	Off	45. 50. 03
Console WM input reset latch	Off	45. 50. 02
Console char control latch	Off	45. 50. 10
Console output error latch	Off	45. 50. 11
WM period latch	Off	45. 50. 11
Console error control	Off	45. 50. 12
Console backspace control	Off	45. 50. 13
Console function control	Off	45. 50. 14
Console printer last column	Off	45. 50. 15
Matrix gate latch	Off	45. 20. 06

Program Reset 5 (12.65.06)

<u>Control</u>	<u>Status</u>	<u>Logic</u>
True latch	Off	16. 20. 14
Complement latch	Off	16. 20. 15
Carry latch	Off	16. 20. 21
No-carry latch	Off	16. 20. 22
Units ctrl latch	Off	16. 30. 02
Body ctrl latch	Off	16. 30. 04
Extension ctrl latch	Off	16. 30. 06
Mplr quot ctrl latch	Off	16. 30. 07
Mult div latch	Off	16. 62. 01
Even-hundreds address latch	Off	14. 71. 40
Reset address mod latch	Off	14. 71. 40
Logic gate special A	On	11. 30. 02
Logic gate triggers (V-Y)	Off	11. 10. 24-27



## Computer Reset

The computer reset key initiates the following operations:

1. Program reset and start reset
2. Resets all check circuits
3. Resets all timing clocks
4. Resets all machine indicators, overflow latches and compare triggers.
5. Inquiry latches are not reset.
6. The tape density latches are never reset.

If the logic clock is not stopped at A when computer reset is pressed, the clock is stored and allowed to advance to logic gate A. After enough time to reset the clock has elapsed, the computer and program resets are initiated.

<u>Signal</u>	<u>Control</u>	<u>Logic</u>
Pwr on + cpr reset	Co Cpr Rst (Key) (or) CE Cpr Rst (Key)	12. 65. 01
Early computer reset (SS)	Pwr on + Cpr Resets	12. 65. 01
Cpr Rst clock start (SS)	Early computer reset	12. 65. 01
Stop latch (off)	Cpr Rst clock start	12. 15. 04

The clock is stopped at A or R.

<u>Signal</u>	<u>Control</u>	<u>Logic</u>
Stop latch (on)	Early computer reset Logic gate A or R	12. 15. 04

After the Cpr Rst Clock St single shot has timed out, computer and program resets are initiated.

<u>Signal</u>	<u>Control</u>	<u>Logic</u>
Computer reset	Pwr On + Cpr Resets Comp Rst Clock St (timed out)	12. 65. 03
All power on + Cpr resets	Pwr On + Cpr Resets Comp Rst Clock St (timed out)	12. 65. 03
Program reset	All power on + Cpr Resets	12. 65. 04

The control, status and logic location of computer reset are:

Computer Reset 1 (12. 65. 03)

<u>Control</u>	<u>Status</u>	<u>Logic</u>
Logic gate A-trigger	Off	11. 10. 10
Logic gate early B-latch	Off	11. 30. 01
Logic gate early F-latch	Off	11. 30. 01
B to last latch	Off	11. 30. 02
Stop latch	On	12. 15. 04
F1 reg not word sep	On	13. 64. 02
F2 reg not word sep latch	On	13. 64. 02
F ch clear latch	On	13. 66. 06
F ch tape call latch	Off	13. 66. 07
Select and R B C on latch	Off	13. 72. 03
F-ch R B C latch	Off	13. 73. 03
E-ch clear latch	On	13. 63. 03
E-ch tape-call latch	Off	13. 71. 04
File ring 7 latch	Off	13. 74. 02
E-ch out mode latch	Off	15. 62. 01
E-ch in mode latch	Off	15. 62. 01
E-ch move mode latch	Off	15. 62. 02
E-ch load mode latch	Off	15. 62. 02
F-ch move latch	Off	15. 63. 02
F-ch interlock	Off	15. 63. 02
Not zero bal latch	On	16. 14. 11
Zero bal latch	Off	16. 14. 12
No overflow latch	On	16. 45. 02
Overflow latch	Off	16. 45. 02
Not div overflow latch	On	16. 45. 01
Div overflow latch	Off	16. 45. 01
High latch	Off	17. 14. 01
Low latch	On	17. 14. 02
Equal latch	Off	17. 14. 03

Computer Reset 2 (12. 65. 03)

<u>Control</u>	<u>Status</u>	<u>Logic</u>
B-data reg reset trigger	Off	39. 10. 01
Y-rd trigger	Off	39. 10. 01
X-rd trigger	Off	39. 10. 01
Z-pulse trigger	Off	39. 10. 02
Y-WR trigger	Off	39. 10. 02
X-WR trigger	Off	39. 10. 02
X-PWR gt ctrl trigger	Off	39. 10. 03
Y-PWR gt ctrl trigger	Off	39. 10. 03
Y-sense strobe trigger	Off	39. 10. 03
Computer reset to tape E-ch	On	12. 65. 03
TAU reset	On	60. 58. 31
In process reset	On	12. 65. 03
E-ch comp reset to file	On	12. 65. 03
Computer reset to tape	On	12. 65. 03
F-chan		
F-ch comp reset to File	On	12. 65. 03
Computer reset to buffer	On	12. 65. 03

## Power-On Reset

The power-on reset is developed through the normally closed points of relay 7 (Section 10.0.00). This causes a power-on reset whenever power sequence is up or down. Program reset, start reset, and computer reset are initiated. The clock is reset to A in the same manner as under Computer Reset. In addition, a system reset is initiated to reset the address registers and the inquiry-request control latch. The system reset is developed as follows:

<u>Signal</u>	<u>Control</u>	<u>Logic</u>
System reset	Pwr Out Cpr Resets Pwr On Reset Comp Rst Clock Start (Timed out)	12.65.02

The following latches are reset with the system reset:

<u>Control</u>	<u>Status</u>	<u>Logic</u>
Inquiry request control	Off	41.30.01
Reset CAR (all latches off)	On	14.71.22
Reset BAR (all latches off)	On	14.71.21
Reset AAR (all latches off)	On	14.71.20
Reset DAR (all latches off)	On	14.71.23
Reset IAR (all latches off)	On	14.71.24
Reset EAR (all latches off)	On	14.71.25
Reset FAR (all latches off)	On	14.71.26

## 2.6.00 INSTRUCTION READ-OUT

Instruction read-out is an operation that reads an instruction word out of core storage and places it into the proper registers to control the execution of the Op code.

Some of the items considered during instruction read-out are:

1. The first character brought out of storage must have a word mark (WM) since this character should be an Op code.
2. The Op code character must be a 1410 Op code or an instruction check stops the operation (1401 Op codes are covered later in the manual).
3. If the Op code is an input-output Op code, the X-field characters must be gated to I/O registers.
4. The Op code determines whether the A-field address is read into the A, B C and D-AR's; the A and C-AR's only; or the C-AR only.
5. There must not be any zones over the units, thousands or ten-thousands positions of an address.
6. Each instruction word must be the correct length.

If an instruction word is ended by a WM at other than an acceptable length, or if there is no WM immediately to the right of the instruction word, an instruction check stops the CPU.

#### 2.6.01 Operation (Figure 2.6-1)

The instruction read-out operation is initiated by setting the I-ring control and I-cycle control latches. The I-ring control latch sets the I-ring to I-ring Op time to identify the character (read out during the first I-cycle) as the Op code.

I-cycle control and I-ring control are set ON by:

1. Program Reset. When the stop latch is reset by the start key the operation starts.
2. Cons Set Start Condition. This initiates the operation when the start key is pressed to start I/E or run mode following a manual operation, such as address set, display and alter. (The process routine latch OFF signifies that the last operation was a manual operation. The latch is ON during I/E or run mode operation.)
3. Last Execute Cycle. This initiates the operation at the completion of the previous instruction.

During the first I-cycle, the address in the IAR is set into the STAR to read out the Op code of the instruction word. (When this operation is initiated immediately after a program reset, the branch-to-00001-latch (reset ON) blocks the IAR from reading out, and generates the address 00001 that is set into the STAR.) When the Op code character is gated onto the B-channel, it must have a WM to set the character into the Op register. The Op register is decoded to control the rest of instruction read-out.

The decoding of the Op register brings up lines called Common Op Code Grouping lines. A Common Op Code Grouping line is a control line for more than one Op code. For example, the 2 Addr Plus Mod Op Codes line identifies the maximum length of the Op codes in that group and is used to control the instruction check circuit.

The Common Op Code Grouping lines consist of three groups: (Figure 2.6-2) instruction read-out, operational, and control.

The instruction read-out group controls the remainder of the I-phase.

The operational and control groups control the E-phase.

If the Op decode is a No Op, the I-ring is not advanced and I-cycle control causes the storage to continue to read out characters until another Op code (WM) is detected. The B-channel WM over the next Op code sets the Op code into the Op register. The instruction read-out operation continues by advancing the I-ring to I-ring 1 time, and the next character is read out of storage.



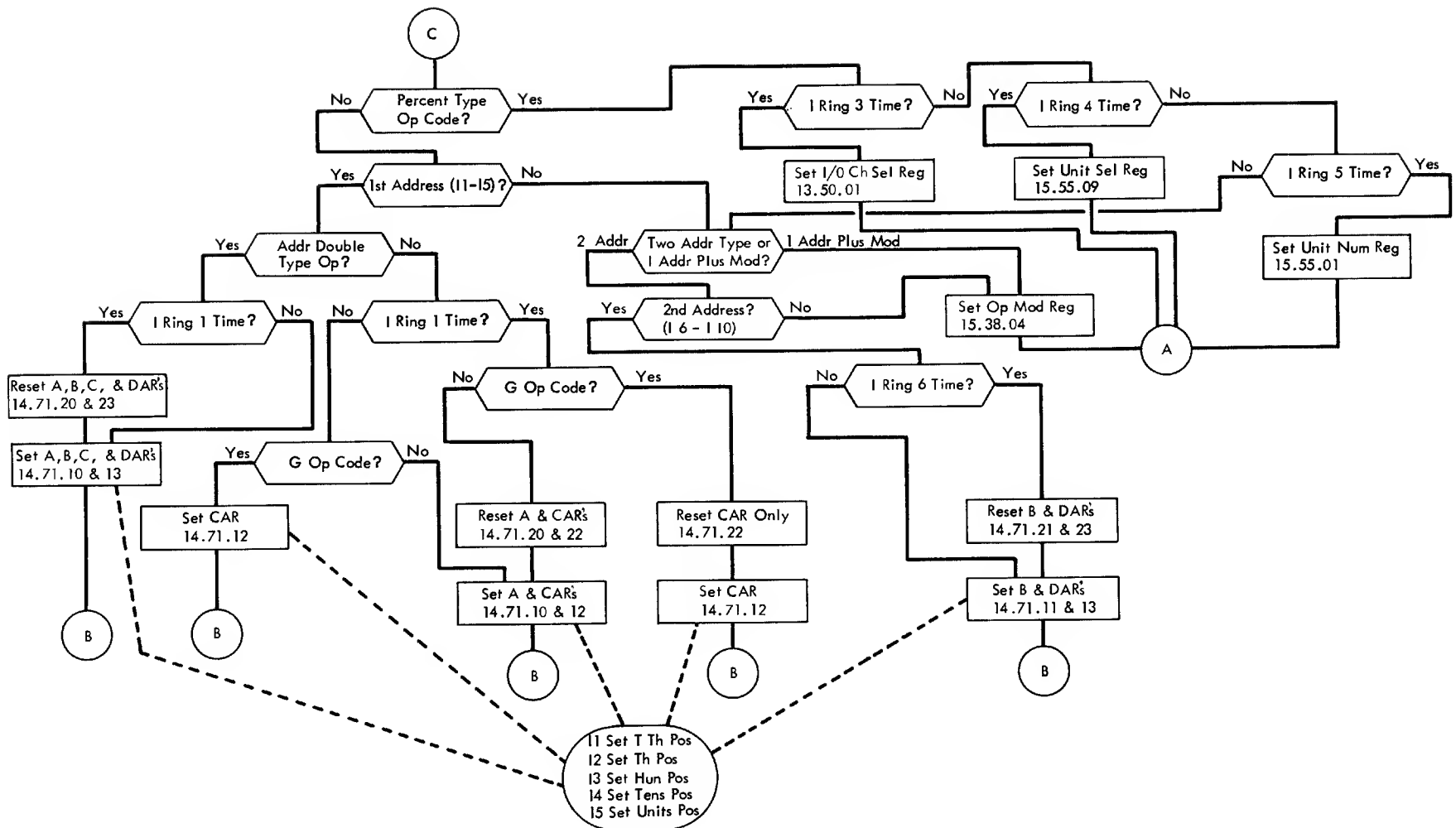


Figure 2.6-1B Instruction Read-Out Operation

# 1410 OP CODES

		COMMON OP CODE GROUPING																?	!	A	S	@	%	E	Z	C	W	V	/	.	,	▯	U	D	J	B	R	X	G	T	M	L	K	F	N
Instruction Read-Out	Percent Type Op Codes																	X																		X	X								
	Not Percent Type Op Codes	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X		X	X	X	X	X		X																				
	Addr Dbl Op Codes	X	X	X	X									X		X	X			X		X	X																						
	Not Addr Dbl Op Codes					X	X	X	X	X	X	X		X				X	X		X				X	X	X	X	X	X															
	1 Addr Plus Mod Op Codes																	X		X		X	X	X																					
	2 Addr No Mod Op Codes	X	X	X	X	X	X	X	X	X				X		X	X																												
	2 Addr Plus Mod Op Codes										X	X							X		X						X	X	X																
	2 Address Op Codes	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X		X		X					X	X	X																		
	Addr Type Op Codes	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X											
	2 Char Only Op Codes																																			X	X								
	C Cycle Op Codes					X	X																			X																			
	No C or D Cy Op Codes							X	X	X	X	X	X	X	X	X		X	X	X																									
	No D Cy at I Ring 6 Ops	X	X	X	X			X	X	X	X	X	X	X	X	X		X		X					X																				
	No Index On 1st Addr Ops																	X							X			X	X																
Operational	Reset Type Op Codes	X	X																																										
	Add or Subt Op Codes			X	X																																								
	Mpy or Div Op Codes					X	X																																						
	Add Type Op Codes	X	X	X	X																																								
	Arith Type Op Codes	X	X	X	X	X	X																																						
	E or Z Op Codes							X	X																																				
	Compare Type Op Codes									X										X					X																				
	Branch Type Op Codes										X	X	X	X						X	X	X	X																						
	No Branch Op Codes	X	X	X	X	X	X	X	X	X					X	X	X	X						X	X	X	X	X	X	X															
	Word Mork Op Codes														X	X																													
M or L Op Codes																												X	X																
Control	1st Scan First Op Codes	X	X	X	X	X	X	X	X	X	X	X	X	X	X				X				X	X																					
	A Cy First Op Codes	X	X	X	X	X	X	X	X					X	X		X							X																					
	Std A Cycle Op Codes	X	X	X	X	X	X	X	X									X								X																			
	B Cy First Op Codes										X	X	X							X																									
	A Reg to A Ch On B Cy Ops	X	X	X	X	X	X	X	X				X	X	X	X		X							X																				
	Op Mod to A Ch On B Cy Ops										X	X					X		X	X	X	X												X	X										
	Load Mem On B Cy Op Codes	X	X	X	X	X	X	X										X																											
	Rgen Mem On B Cy Op Codes									X	X	X		X					X	X	X	X		X									X	X											
	Stop of F on B Cy Op Codes												X	X					X		X	X																							
	Stop at H on B Cy Ops										X	X								X																									
	Stop of J on B Cy Op Codes	X	X			X	X	X	X										X																										
	RO B AR On Scan B Cy Ops							X	X	X	X	X	X	X	X	X		X	X	X	X	X		X																					
	RO A AR On A Cy Ops	X	X	X	X			X	X	X					X	X		X																											

Figure 2.6-2 IBM 1410 Common Op-Code Grouping

If the Op decode is neither an N or G, one of three common Op-code grouping lines must come up. They are percent type, not-percent type, or two-character-only Op codes.

### Two-Character-Only Op Codes

For both two-character-only Op codes, the character at I-Ring 1 Time must be the Op modifier, and it cannot have a word mark over it or an instruction check results. The Op modifier character is stored in the Op modifier register and the I-ring is advanced to I-Ring 2 Time where the next character is read out to insure that there is a word mark over it.

### Not Percent Type Op Codes

All of these Op codes (except R and X) are the only Op codes that can be chained. Three more common Op-code grouping lines identify the maximum length of these Op codes. The lines are: One Address Plus Mod Op codes, 2 Addr No Mod Op codes, and 2 Addr Plus Mod Op codes. They insure that there are word marks at I7, I11, and I12 time. I-ring 1-5 Times identify the A-field address and I6-8 identify the B-field address for Two Address Type Op codes.

### Percent-Type Op Codes

All move and load Op codes and the unit control Op code fall into this group. Move and load Op codes also bring up 2 Addr Plus Mod Op code while the U-Op code conditions One Address Plus Mod Op codes.

When the percent-type Op-code line comes up, it causes two quick advance pulses that advance the I-ring to I3 during the second I-cycle.

I cy	I cy			I cy	I cy	I cy	I cy
I op	I1	I2	I3	I4	I5	I6	I7

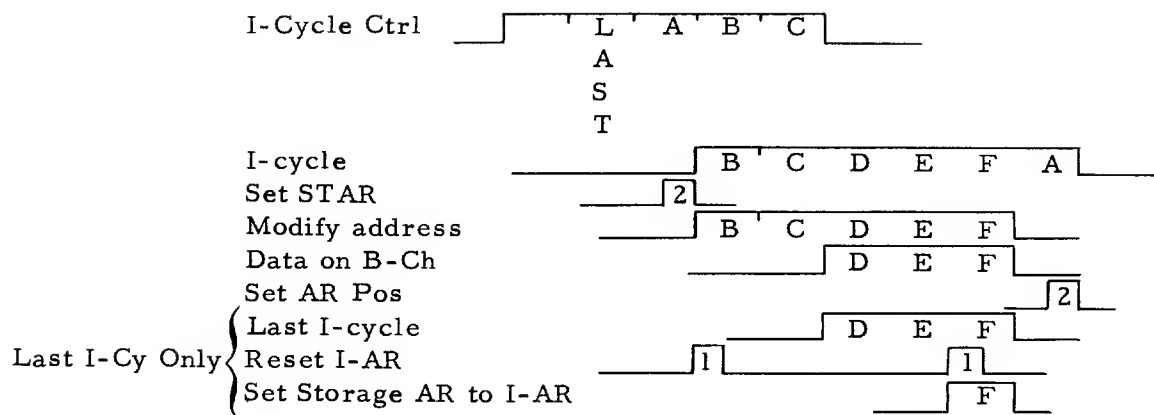
The character at I3 time is the I/O channel-select character. It is set into the I/O channel-select register. The character at I3 is the unit-select character, and it is set into either the E- or F-units select register depending on the I/O Ch Set Register. At I5 the character is set into the E- or F-unit number register. The character at I6 is set into the Op modifier register for the One Address Plus Mod Op codes, or it is set into the ten-thousands position of the B- and D-AR's for the 2 Addr Plus Mod Op codes. (When the command is overlapped, the B-field address is also set into the E- or F-AR depending on the channel-select register.)

### I-Cycle

The storage cycles during instruction read-out are 4.5 microseconds long (stop at F). As each character is read out into the B-data register, it is regenerated.

The STAR is set on 2nd CP of LGA. From LGB to LGF the address in STAR is modified by plus one (controlled by I-cycle control) and read back into the I-AR.





The character is available on the B-channel around LGD. The I-cycle latch gates the character to the A-data register and to the A-channel. I-cycle also controls the assembly to Use A-Ch WM, zones, and Num.

On the last I-cycle of any instruction read-out operation, there must be a WM, if the instruction is of correct length. Because this character is usually the Op code of the next instruction, it must be read out again. As this character is read out, the address in STAR is being modified by plus-one to be read into the IAR. When the WM identifies the cycle as the last I-cycle (around LGD), the gate to set the IAR from the address channel is dropped, and the IAR is reset (1st CP, LGF). Set Mem AR to IAR is brought up at LGF to gate the address of the next Op code, still in STAR, to the IAR.

In order to satisfy the address-channel validity check during I-phase, zeros must be inserted on the channel at LGA of I Op, Op Mod character, and last instruction read-out cycle. All address characters and X-control-field characters are gated onto the address channel. The %, @ , and \* characters (8-4 Bits) are translated to a four.

## 2.6.02 Chaining

There are two items that effect an Op code, when the Op code is chained. They are:

1. Some two-address-type Op codes, when chained to use a single address, use this address as both the A and B-field addresses. These Op codes are called address double-type Op codes. Not-Address Double-Type Op code uses the contents remaining in the BAR at the completion of the previous E-phase for the B-field address.
2. Some Op codes require the A and C, or B and D-address registers to contain the same address before E-phase can start.

## Address Double-Type Op Codes

The A-field of an address double type Op code is read into the A, B, C, and D-ARs so that if a WM is read out at I Ring 6 Time the single address is used as both the A- and B-field addresses.

The A-field of not-address double-type Op codes is read into the A- and C-ARs only.

### Arithmetic Type Op Codes

If an arithmetic type Op code is chained at I Ring 1 Time, the CPU takes a D-Cycle during which the address in the BAR is set into STAR, modified by zero, and read into the D-AR (the B- and D-AR's must contain the same address). Chaining multiply, divide, and T Op codes at I Ring 1 Time initiates a C-cycle to update the C-AR.

#### 2.6.03 Op Register Set Check

The Op register set check insures that the Op register receives a set impulse during I-Op time. The conditions that cause Set Op Reg also bring up check Op register set (Figure 2.6-3). Set Op Reg and Check Op Register Set impulse two triggers that must be both ON or both OFF at error sample time, or an Op register set error stops the CPU.

#### 2.6.04 Op-Modifier Register Check

The Op-modifier register check insures that the Op modifier register receives a set impulse at Op-modifier character time for Op codes with modifiers. The conditions that cause Set Op Mod Reg also bring up Check Op Mod Set (Figure 2.6-4). Set Op Mod Reg and Check Op Mod Set impulse two triggers that must be both ON or both OFF at error sample time or an Op Mod Reg Error stops the CPU.

#### 2.6.05 Instruction Check

The instruction check circuit detects the following programming errors:

1. Addressing an Op code without a word mark.
2. Using an undefined Op code or a 1401 Op code when not in the 1401 mode.
3. Using an instruction word that is not a proper length (proper lengths are shown in Figure 2.6-5).
4. Failure to have a word mark immediately to the right of the instruction word.
5. A table search Op with 8, A, or B-bits in the d modifier.

The circuit also detects these system failures:

1. Failure to set a cycle-control latch for the next cycle.
2. Failure to control the length of the storage cycle.

#### Operation (Figure 2.6-6)

During every cycle, conditions must be met to set the cycle-control latch for the next cycle. If none of the cycle-control latches are set, the instruction check gate is conditioned, and at error sample time the CPU is stopped. During every I-cycle the WM bit on the B-channel is used to condition Last Insn RO cycle or the cycle-control latch for the next cycle. (Last Insn RO cycle is usually used to condition the cycle-control latch for the first cycle of E-phase.)

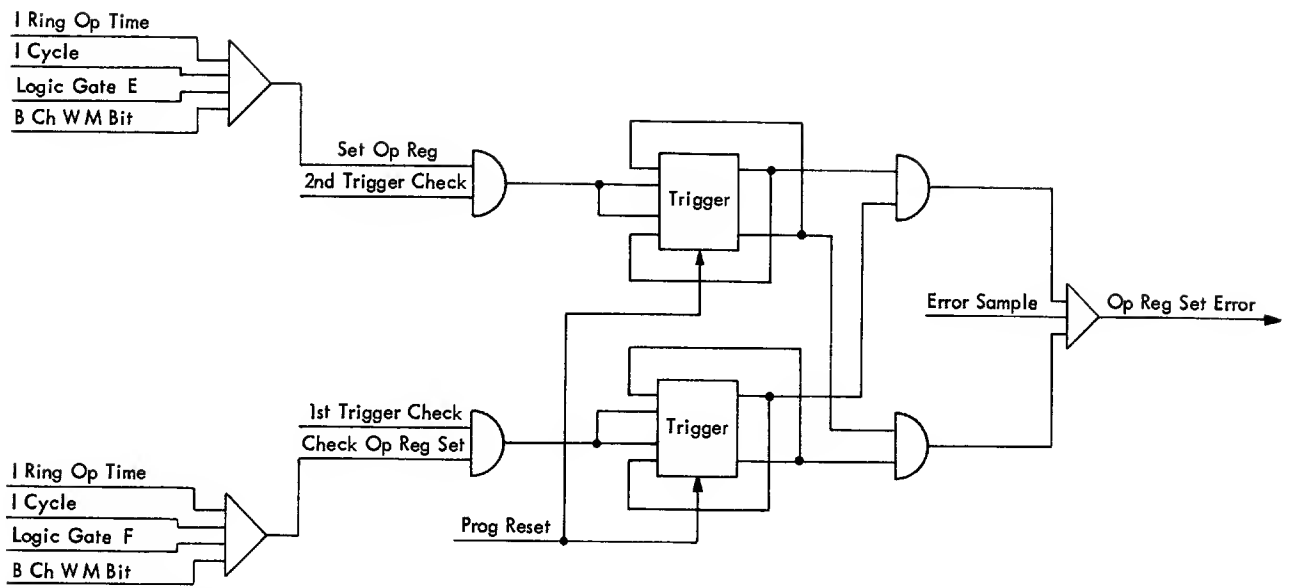


Figure 2.6-3 Op Register Set Check

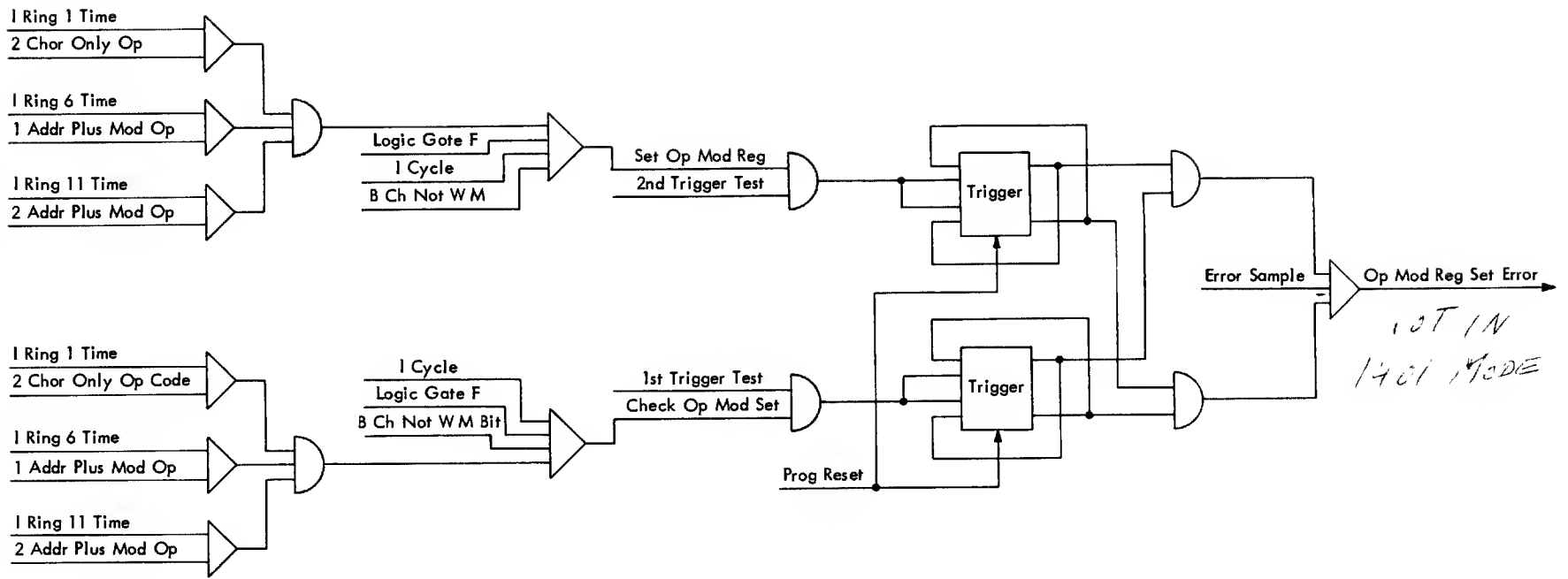


Figure 2.6-4 Op Mod Register Set Check

OP CODE/INSTR	FUNCTION	ACCEPTABLE LENGTHS		
A (A) (B)	Add	1	6	11
S (A) (B)	Subtract	1	6	11
? (A) (B)	Zero and Add	1	6	11
I (A) (B)	Zero and Subtract	1	6	11
@ (A) (B)	Multiply	1	6	11
% (A) (B)	Divide	1	6	11
J (I) d	Branch if Internal Ind On	1	7	
R (I) d	Branch if I/O Status Ind On (Ch 1)		7	
X (I) d	Branch if I/O Status Ind On (Ch 2)		7	
B (I) (B) d	Branch if Char Equal	1	6	12
W (I) (B) d	Branch if Bit Equal	1	6	12
V (I) (B) d	Branch on WM and/or Zone	1	6	12
D (A) (B) d	Move Data	1	6	12
Z (A) (B)	Move Char and Suppr Zeros	1	6	11
E (A) (B)	Move Char and Edit	1	6	11
C (A) (B)	Compare	1	6	11
T (A) (B) d	Table Lookup	1	6	12
G (C) d	Store Addr Reg		7	
, (A) (B)	Set Word Mark	1	6	11
(A) (B)	Clear Word Mark	1	6	11
/ (I) (B)	Clear Storage and Branch	1	6	11
. (I)	Halt and Branch	1	6	
N	No Op	-	-	-
M (X) (B) R/W	Read or Write without WM			12
L (X) (B) R/W	Read or Write with WM			12
U (X) d	Control Unit		7	
K d	Stacker Sel and Feed	2		
F d	Control Carriage	2		

Figure 2.6-5 Op Code Lengths

Set I Cycle Control (12.13.02)

B Ch WM	I Op		Not 1401 Mode
			Set I Cy Ctrl No Op
			ARS No Op • I Cycle
			Op Mod Time • Not 1401
B Ch Not WM	I 1 + 2 + 3 + 4	I Cy	Address Type Op Codes
	I 5	C Cy	
B Ch Not WM	I 5 + 10		Index Not Required
B Ch Not WM	I 6 + 7 + 8 + 9	I Cy	Two Address Type Op Codes
	I 10	D Cy	

Last Insn RO Cycle (12.13.05)

	I 1	C Cy	C Cycle Op Codes
	I 1	D Cy	Add Type Op Codes
B Ch WM	I 1		No C or D Cycle Op Codes
B Ch WM	I 2		2 Chor Only Op Codes
	I 6	D Cy	Mult + Divide Op Codes
B Ch WM	I 6		No D Cy or I 6 Op Codes
B Ch WM	I 7		1 Addr Plus Mod Op Codes
B Ch WM	I 11		2 Addr No Mod Op Codes
B Ch WM	I 12		2 Addr Plus Mod Op Codes

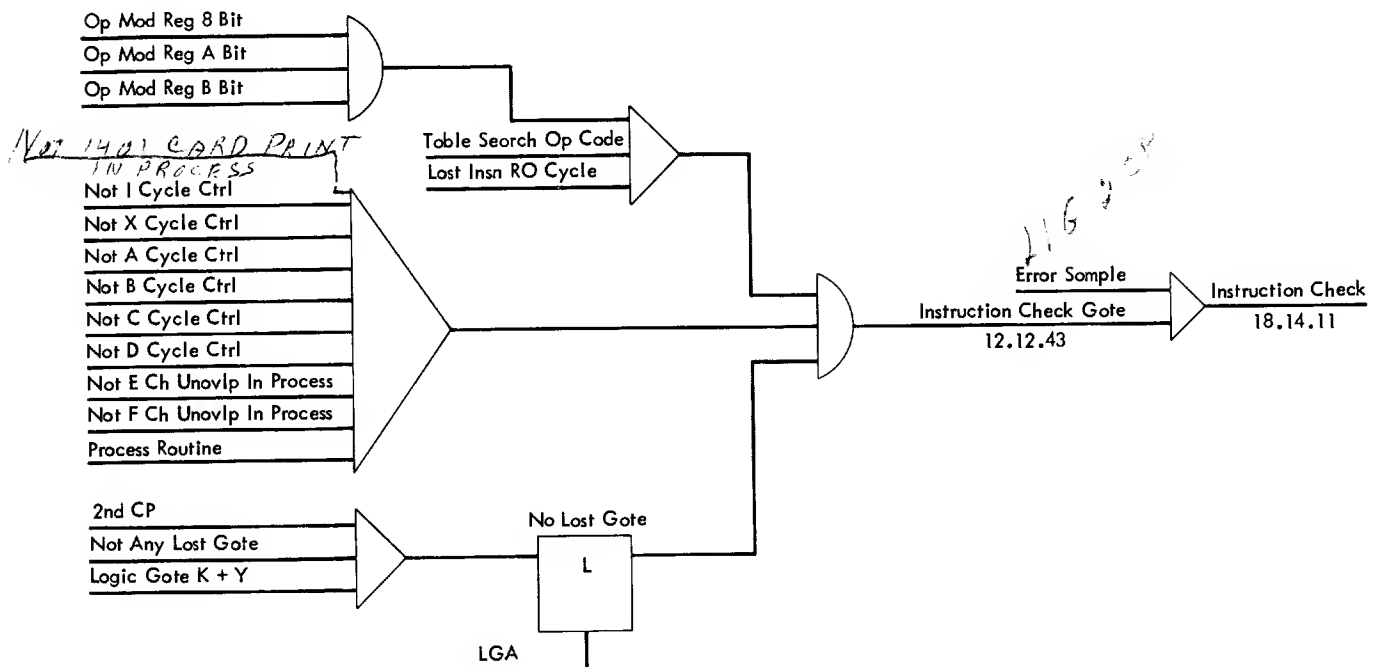


Figure 2.6-6 Instruction Check

For example, there must be a B-Ch WM at I-ring Op time to set the I-Cycle Ctrl latch for the next character.

If the programmer uses an undefined Op code, the Op-register decode does not condition any common Op-code grouping lines, that are necessary to set the cycle-control latch for the I-cycle at I-ring 1 time.

Since I/O commands can be overlapped the E- and F-cycle-control latches cannot be used in the check circuit. When the commands are overlapped the I cycle-control latch is set during last instruction RO cycle. When the command is unoverlapped Not E and Not F Ch Unovlop In Process (set by status sample A that is conditioned by a B-Ch WM) prevent an instruction check. The process routine line prevents an error during manual operations.

Any transistor failure that prevents a cycle-control latch from setting is detected by this check circuit.

The instruction check gate is also conditioned by the no-last-gate latch. This latch is set ON, if either logic-clock ring is advanced to last position without getting a last logic gate. Without the check circuit, the next clock pulse would advance the clock off the end of ring.

#### 2.6.06 Address Check

The address-check circuit insures that there are no zones over the units, thousands, and ten-thousands positions of an address field in an instruction word.

If the assembly channel contains an A- or B-bit at I-ring 1, 2, 5, 6, 7, or 10 time of an I-cycle, the address latch is set ON and the CPU stops at error sample time (Figure 2.6-7). The Not Op Mod Character Time prevents an address check on the modifier character read out at I-ring 1 or 6 time.

The address-check circuit also insures that during a scan the address is not modified past the last or first position of storage (wrap-around condition).

The wrap-around condition varies with the size of the storage.

1. On a 10K machine the Mod By Plus One latch that is still ON at logic gate F (ten-thousands position being modified) indicates that the address is 9,999. The address-check latch is turned ON and the CPU is stopped at error sample time.
2. Also on a 10K machine the Mod by Minus One latch that is still ON at logic gate F indicates that the address is 0,000.
3. On a 20K machine, a 0-bit and a 2-bit on the address channel at logic gate F indicate the address is modified from 19,999.
4. On a 40K machine, a 0-bit and a 4-bit on the address channel at logic gate F indicate the address modified is 39,999.
5. On either the 20 or 40K machine, a 0-bit and a 8-bit on the address channel at logic gate F indicate the address modified was 00,0000.

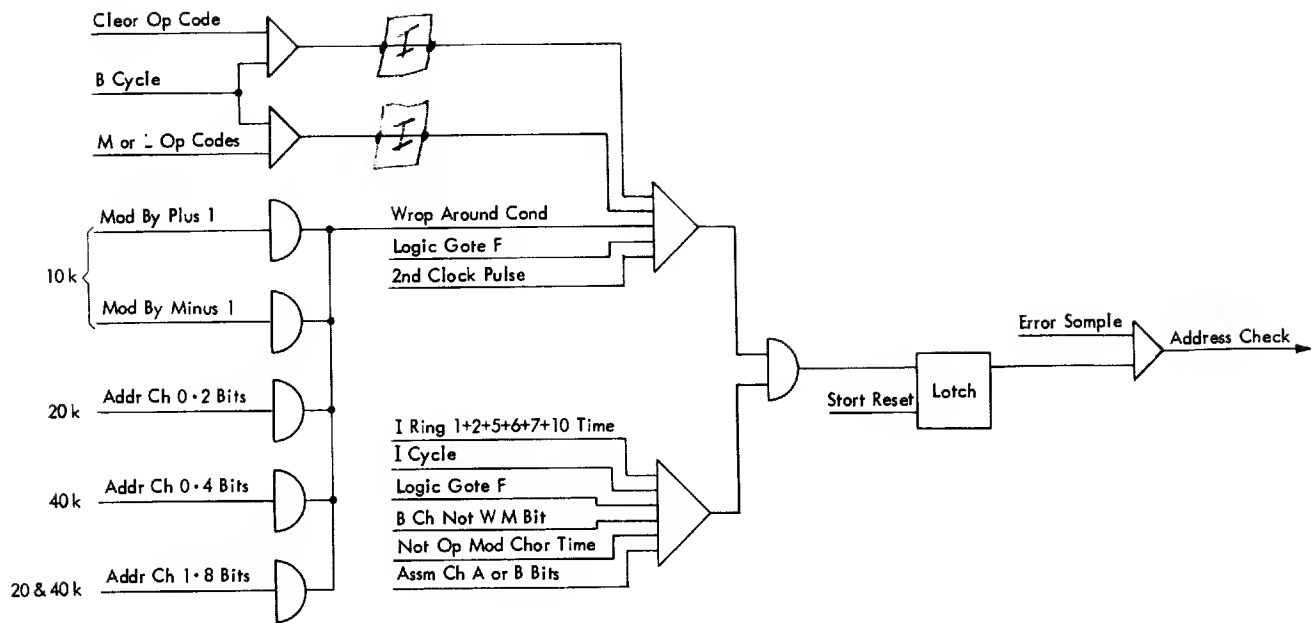


Figure 2.6-7 Address Check



A wrap-around condition during a move and load Op code and a clear Op code does not cause an error.

The M and L Op codes are always stopped at the last position of storage.

## 2.7.00 INDEXING

Indexing, which is a standard feature in the IBM 1410, uses fifteen five-digit indexing fields in core storage. Any address of any instruction (except a G Op code) can be indexed. When an address is to be indexed, the contents of the specified index register adds algebraically to that address, immediately after the units position reads out of storage (Figure 2.6-1). Neither the contents of the index register nor the address originally stored in core storage as part of the instruction word is affected by the operation.

For this operation, the address in the instruction is considered to be positive regardless of the sign indication it may have over the units position. When the index register has a positive sign, it causes its contents to add to the address in the instruction word. When the index register has a negative sign, it causes the contents to subtract from the address in the instruction word. This modification of the address must result in a valid core-storage address, or when an attempt is made to use this address, an error will result.

### 2.7.01 Index Register Locations

Seventy-five characters of core storage are assigned for use as index registers. The locations are:

<u>Index Register</u>	<u>Locations</u>	
	<u>From</u>	<u>To</u>
1	00025	00029
2	30	34
3	35	39
4	40	44
5	45	49
6	50	54
7	55	59
8	60	64
9	65	69
10	70	74
11	75	79
12	80	84
13	85	89
14	90	94
15	95	99

These locations in core storage can be addressed the same as other locations, and are available as general storage, if not required for indexing. Word marks can be located in this area for the convenience of other operations because they do not affect the result of indexing and are not destroyed.

### 2.7.02 Index Tags

The zone bits over the tens and hundreds position of the address in the instruction are used as tags to specify whether the address is to be indexed, and if so which index register to use.

The following table explains the significance of these tag bits.

SIGNIFICANCE OF TAG BITS

<u>Hundreds Position B-Bit</u>	<u>Hundreds Position A-Bit</u>	<u>Tens Position B-Bit</u>	<u>Tens Position A-Bit</u>	<u>Index with Register</u>
8	4	2	1	
			X	1
		X		2
		X	X	3
	X			4
	X		X	5
	X	X		6
	X	X	X	7
X				8
X			X	9
X		X		10
X		X	X	11
X	X			12
X	X		X	13
X	X	X		14
X	X	X	X	15

### 2.7.03 Operation (Figure 2.7-1)

As the hundreds and tens positions of the addresses are read out during instruction read-out, the zones from the assembly area set the index-tag latches (Figure 2.6-1). There are A, B, and C-tag latches for each of the two positions of index tags, that correspond to the A, B or zone C-bits (the zone C-bit is generated in the A-channel validity-check circuit in order to make the number of zone-bits odd).

If the index-tag latches indicate that there are no index tags, Index Not Required allows instruction read-out to continue.

When the index-tag latches indicate that there is a tag, an Index Required line:

1. stops the I-ring from advancing, after the units position of the address is read into the address register (I-ring 5 or 10 time).
2. causes the machine to take five X-cycles.

During the five X-cycles, the five positions of the index register read out of storage onto the B-channel. The five digits add units position first, to the address

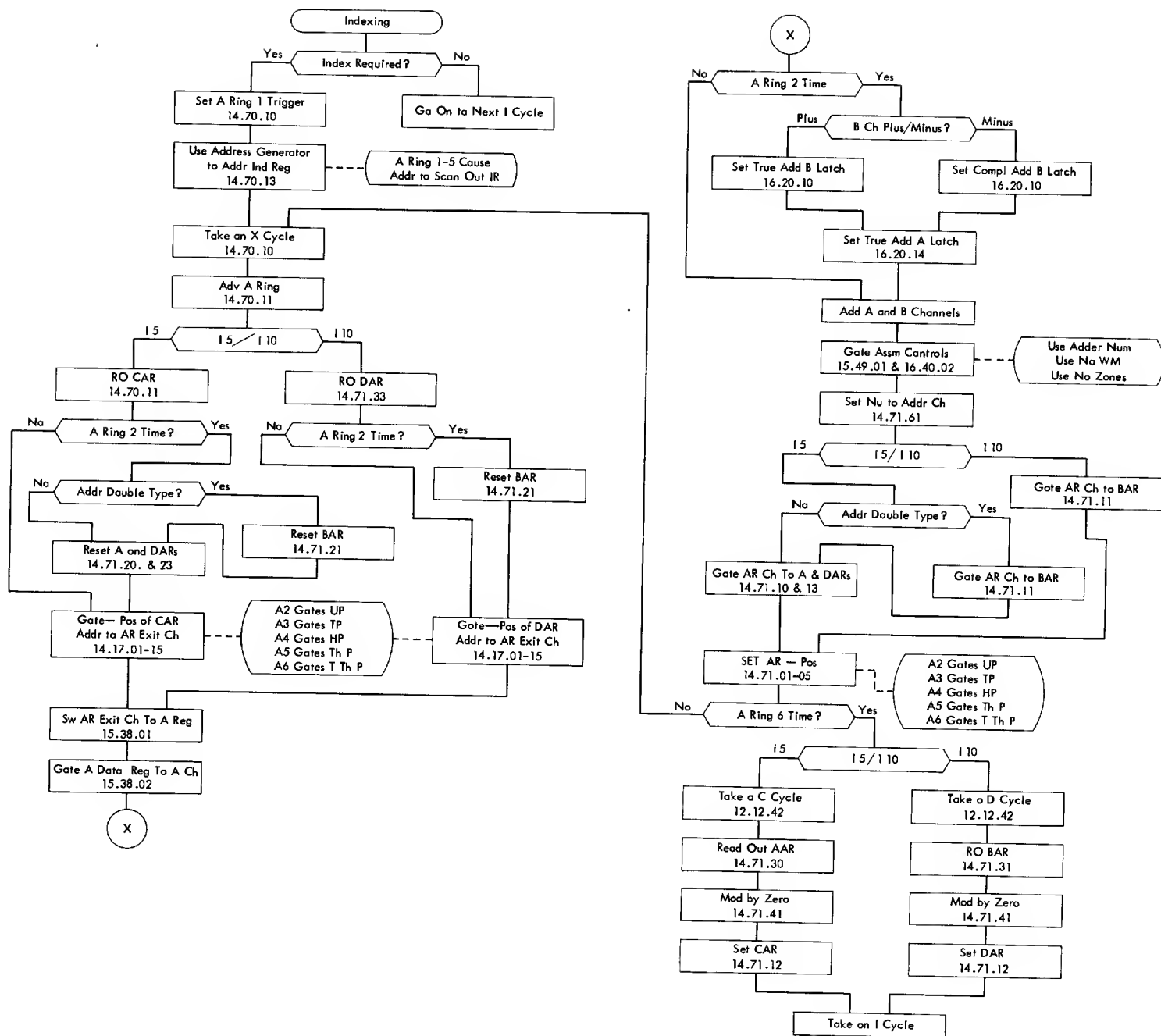


Figure 2.7-1 Indexing

in either the C or D-ARs, depending on which field is being indexed (I5 index A-field, I10 index B-field). A six-position ring called the A-ring is used to identify the five positions of the address.

The A-ring-1-time trigger is set at the end of the I-cycle at I-ring 5 or 10 time, when indexing is required.

A-ring 1 time and the index tag latches are combined to generate the address of the units position of the selected index register. At LGB of the first X-cycle, the A-ring advances to A-ring 2 time.

At A-ring 2 time, the units position of the address bus is gated to the A-data register (the address bus contains the address in the C- or D-AR). From the A-data register the character is gated to the adder via the A-channel, where it is combined with the units position of the index register that is read out of storage onto the B-channel. (The sign over the units position of the I R determines whether the B-side of the adder is true- or complement-added.)

The result from the adder is gated through the assembly to the address channel where it is set into the A- or B-AR. If the Op code is an address-double-type Op code the indexed A-field address reads into the A, B and D-ARs in case there is no B-field address.

A-ring 2 and the index tag latches are combined to generate the address of the tens position of the index register. The CPU takes another X-cycle and advances the ring to A3 time and the tens position is added and stored the same as the units position.

When the A-ring is at A6, the last position is combined. At the end of the X-cycle the CPU takes a C or D-cycle, depending on the address being indexed in order to update the C or D-AR with the new indexed address. During this operation the I-ring is not advanced and at the completion of the C- or D-cycle the CPU goes back to taking I-cycles to complete instruction read-out.

## 2. 8. 00 DATA HANDLING OP CODES

The common Op-code grouping lines (Figure 2. 6-2) are also used to control E-phase. The operational group of lines is used by similar types of Op codes. The control group of lines is used by any type of Op codes that require the same controls. The function of these lines is shown in Figure 2. 8-1.

### 2. 8. 01 Move Data Op Code (D)

#### Op Code Function

The D Op code causes data in storage to be moved from the A-field address to the B-field address.

#### Op Modifier Function

The 1, 2, and 4-bits of the Op modifier determine which portion of the A-field characters transfers.

<u>Common Op Code Grouping Lines</u>	<u>Function</u>	<u>Time</u>	<u>Logic</u>
1st Scan First Op Codes	Set 1st Scan Ctrl	Last Insn RO Cycle	12. 30. 05
A Cy First Op Codes	Set A Cycle Ctrl	Last Insn RO Cycle	12. 12. 41
Std A Cycle Op Codes	Set B Cycle Ctrl	A Cycle	12. 12. 44
	Stop at F	A Cycle	12. 12. 30
	Rgen Memory	A Cycle	12. 50. 02
	Use A Ch WM	A Cycle	15. 49. 02
	Use A Ch Zones	A Cycle	15. 49. 03
	Use A Ch Nu	A Cycle	15. 49. 03
	Rgen Unit Body Ctrl	A Cycle	16. 30. 01
	Rgen Ext Ctrl	A Cycle	16. 30. 05
	Rgen 1st Scan Ctrl	A Cycle	12. 30. 05
	Rgen 2nd 3rd Scan Ctrl	A Cycle	12. 30. 06
B Cy First Op Codes	Set B Cycle Ctrl	Last Insn RO Cycle	12. 12. 44
A Reg to A Ch on B Cy Ops	Gate A Data Reg to A Ch	B Cycle	15. 38. 02
Op Mod to A Ch on B Cy Ops	Gate Op Mod Reg to A Ch	B Cycle	15. 38. 02
Load Mem on B Cy Op Codes	Load Mem	B Cycle	12. 50. 01
Regen Mem on B Cy Op Codes	Rgen Mem	B Cycle	12. 50. 02
Stop at F on B Cy Op Codes	Stop at F	B Cycle	12. 12. 30
Stop at H on B Cycle Ops	Stop at H	B Cycle	12. 12. 32
Stop at J on B Cy Op Codes	Stop at J	B Cycle	12. 12. 32
RO B AR on Scan B Cy Ops	RO B AR	B Cycle Ctrl, 1st + 2nd + 3rd Scan Ctrl	14. 71. 31
Read Out A AR on A Cy Ops	RO A AR	A Cycle Ctrl	14. 71. 30

Figure 2.8-1 Common Op Code Grouping Lines

<u>Op Mod Bit</u>	<u>Use</u>
1	A-Field - Num
Not 1	B-Field - Num
2	A-Field - Zones
Not 2	B-Field - Zones
4	A-Field WM
Not 4	B-Field WM

The 8-bit determines the direction of the scan.

8-Bit	Transfer high to low order
Not 8-Bit	Transfer low to high order

The A- and B-bits, along with the 8-bit, determine when the data transfer should be ended.

<u>Op Mod Bits</u>	<u>Stop at</u>
$\overline{8}, \underline{A}, \overline{B}$	1st WM - A-Field
$\overline{8}, \underline{A}, B$	1st WM - B-Field
$\overline{8}, \underline{A}, \overline{B}$	1st WM A or B-Field
$\overline{8}, \underline{A}, \overline{B}$	1st WM - A or B-Field
$\overline{8}, \underline{A}, \overline{B}$	1st RM - A-Field
$\overline{8}, \underline{A}, B$	1st GM, WM. A-Field
$\overline{8}, \underline{A}, \overline{B}$	1st RM or GM, WM - A-Field
$\overline{8}, \underline{A}, B$	After one position

Op-code grouping lines:

Instruction Read-out

Not percent type  
 No addr double  
 Two addr plus mod  
 Two addr  
 Address type  
 No C- or D-cycle  
 No D-cycle at I6

Operational

No branch

Control

A-cycle first  
 Standard A-cycle  
 A-reg to A-ch on B-cycle  
 Load mem on B-cycle  
 Stop at J on B-cycle  
 Read out BAR on scan B-cycle  
 Read out AAR on A-cycle

## Operation

During the Last Insn RO cycle, the common Op-code grouping lines set up the controls to take a standard A-cycle first. During this A-cycle the AAR is used to read out the first position of the A-field.

The 8-bit position of the Op modifier register must be analyzed to determine the direction of the scanning of the two fields. If there is no 8-bit in the Op modifier register during Last Insn RO cycle, the first Scan Ctrl latch is set (12. 30. 05). If there is an 8-bit, the 2nd Scan Ctrl latch is set (12. 30. 06).

During the A-cycle, the first character is regenerated in storage, and also set into the A-data register (15. 38. 01). Standard A-cycle Ops combined with an A-cycle causes the next cycle to be a B-cycle. During this B-cycle the BAR is read out to address the first position of the B-field in storage.

Also during this B-cycle, the 1, 2, and 4-bits of the Op modifier register control the assembly (Figure 2. 8-2) in order to combine the portion or portions of the A-field character on the A-channel with the B-field character on the B-channel.

At the end of the B-cycle the Op modifier register 8, A and B bits are analyzed to determine whether the transfer is completed (last execute cycle) or whether another A-cycle should be taken to transfer the next position of the A-field (Figure 2. 8-3).

### 2. 8. 02 Move Character and Suppress Zero Op Code (Z)

#### Op-Code Function

The Z-Op code causes data in storage to move from the A-field to the B-field address. All zeros and commas to the left of the first significant digit are replaced by blanks, and the sign is removed from over the units position.

Op-code grouping lines:

	<u>Instruction Read-out</u>
	Not percent type
	Not addr double
	Two addr no mod
	Two addr
	Address type
	No C- or D-cycle
	No D-cycle at I6
	<u>Operational</u>
	E or Z
	No branch
	<u>Control</u>
	1st scan first
	A-cycle first
	Standard A-cycle

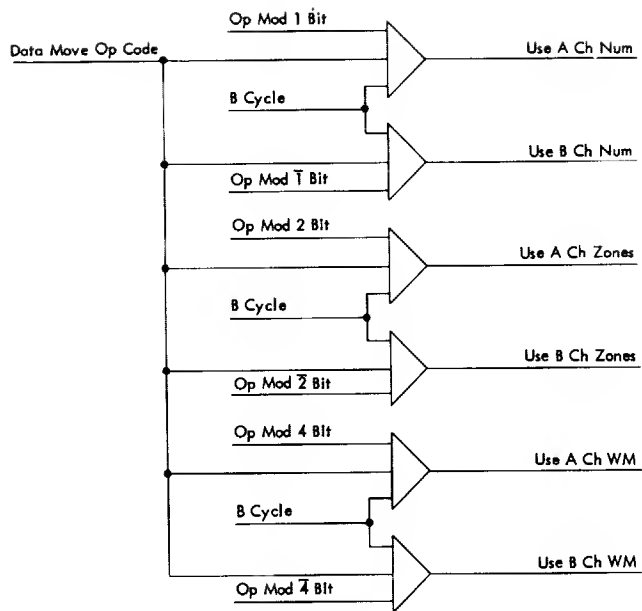


Figure 2.8-2 Assembly Unit Controls

OP MOD REGISTER								
	A	B	AB	B	8A	8B	8AB	Blank
A Ch WM	X		X	X	X		X	X
A Ch RM								X
A Ch GM, WM		X	X	X		X	X	X
B Ch WM								X
Any Character								X

X Indicates Data Move Take A Cycle down (12.12.40)

Also- Data Move Last Execute Up (12.12.50)

Figure 2.8-3 Data-Move Take A-Cycle



A-reg to A-ch on B-cycle  
Load mem on B-cycle  
Stop at J on B-cycle  
Read out BAR on scan B-cycle  
Read out AAR on A-cycle

#### Operation (Figure 2.8-4)

This Op code requires two scans. First Scan First sets the 1st Scan Ctrl latch at the end of the Last Insn RO cycle. During this forward scan the A-field characters transfer to the B-field.

The units latch, which is set ON during Last Insn RO cycle, identifies the first A-field character as the units position. The E-phase starts with an A-cycle first, using the AAR to read out the first A-field character that is stored in the A-data register. Because this is a standard A-cycle Op code, the units and 1st Scan latches are regenerated while the B-Cycle Ctrl latch is set. During this B-cycle, the numerical portion of the A-field character is gated through the assembly from the A-data register via the A-channel. The units latch, which is set ON during this first B-cycle, controls the assembly to Use No Zone (eliminates the A-field sign) and Write a WM. This WM is used to identify the end of the B-field on the reverse scan. The zero suppress latch is also set during this B-cycle although it is not used to suppress zeros until the second scan.

At the end of the B-cycle, the A-Ch WM bit is analyzed to determine whether another A-cycle should be taken or whether the data transfer is complete and the second scan should be started.

If there is not A-Ch WM the 1st Scan latch is regenerated and the body latch is set to control the assembly to Use A-Ch Nu, Zones, and No WMs for the next B-cycle.

When there is an A-Ch WM, the 2nd Scan latch is set to control the address modifier to Modify by Plus One. During the last B-cycle of the first scan, the address in the BAR is modified by minus one. Since no other address register contains the address of the end of the B-field, the BAR must be used during the reverse scanning of the field. When the BAR is used to address storage during the first B-cycle of the second scan, the character that is addressed is the position to the left of the B-field. This character must be read back into storage. If there is a WM over the character, it must not be allowed to stop the second scan. This B-cycle is called an edit skid cycle. It is identified by setting the MQ latch. The edit skid cycle controls the full B-channel character through assembly and back to storage.

The extension latch is set and another B-cycle is taken. During this B-cycle the high-order position of the B-field reads out onto the B-channel. When this B-channel character is zero, comma, or blank; the assembly is controlled to write a blank because the zero-suppress latch is still ON. When the B-channel character is a significant digit (digits 1-9), the zero-suppress latch is reset and all B-channel characters from that point on are gated back to storage.

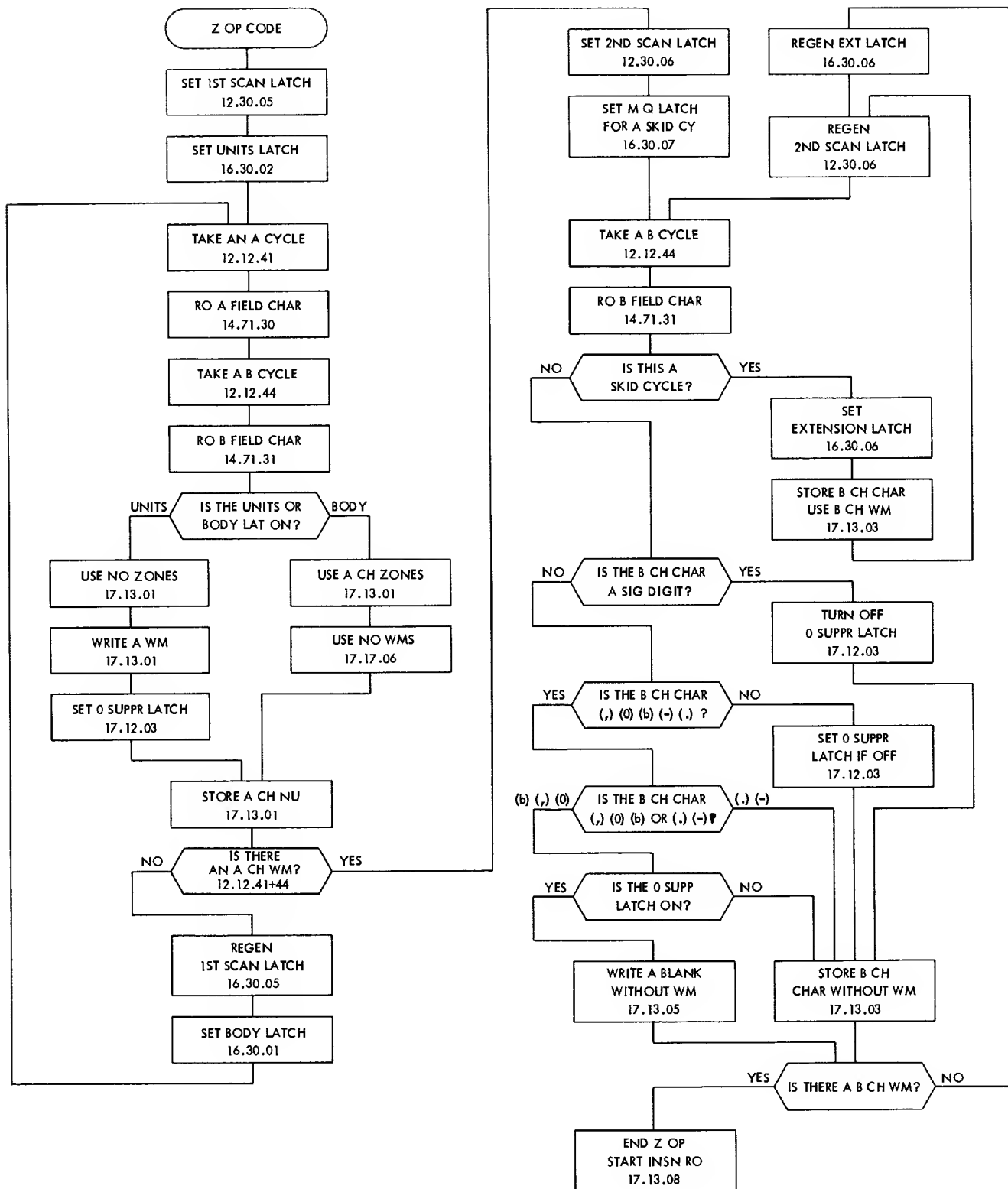


Figure 2.8-4 Move Character and Suppress Zeros

When the B-channel character contains a WM, Last Execute Cycle is conditioned to end the Z-Op and initiate instruction read-out.

There is one more item to consider for this Op code: multiple field transfer. For example:

v  
A-field    0010b@00.25

During the second scan, the zeros are suppressed up to the 1, where the zero-suppress latch is turned off. The zero and blank are stored back into storage unchanged. When the @ is read out, it is recognized not as a significant digit and not as a zero, blank, comma, decimal or minus sign. This causes the zero-suppress latch to come back ON to suppress the high-order zeros of the next field.

Result B-field    10 @ .25

### 2.8.03 Compare Op Code (C)

#### Op-Code Function

The C-Op code causes the A-field to compare with the B-field to determine whether the B-field is greater than, equal to, or smaller than the A-field. When the A-field is shorter than the B-field, the high latch is set ON to indicate that the B-field is greater than the A-field regardless of their values.

Op-code grouping lines:

Instruction Read-out  
Not percent type op codes  
Not addr double  
2 addr no mod  
Two addr  
Addr type  
No C- or D-cycle  
No D-cy at I ring 6

Operational  
Compare type  
No branch

Control  
1st scan first  
A-cycle first  
Standard A-cycle  
A-reg to A-ch on B-cycle  
Regen Mem on B-cycle  
Read out BAR on scan B-cycle  
Read out AAR on A-cycle

### Operation (Figure 2.8-5)

During Last Insn RO cycle the units, 1st Scan, and A-Cycle Ctrl latches are set. The AAR is used to read out the first A-field character and to store it in the A-data register. During the B-cycle that follows the standard A-cycle, the first B-field character reads out onto the B-channel, while the A-data register is gated to the A-channel. The characters on the channels are compared in the compare and adder units (Figure 2.8-6).

The output of the compare matrix sets the high, low, or equal latches. The equal latch is set ON only for the units position of the fields, as defined by the units latch. The operation continues (A-cycle, B-cycle, A-cycle, etc.) to compare the fields and to reset the low and high latches as required.

When a B-channel WM reads out, the operation terminates, and the last latch that is set ON remains ON for the programmer to test later in his program.

Example:

B-field	6 0 2 1 5
A-field	8 9 2 0 5
B-channel	5 1 2 0 6
A-channel	5 0 2 9 8
Compare latch set	E H H L L*

The low latch remains set.

B-field	0 0 2 1 5
A-field	9 2 1 5
B-channel	5 1 2 0
A-channel	5 1 2 9
Compare latch set	E E E L H*

The high latch is set, because the B-field is longer than A.

#### 2.8.04 Edit (E)

The E-Op code transfers data in storage from the A-field location to a control word in the B-field location where desired punctuation is inserted. Besides supplying the punctuation, the control field controls:

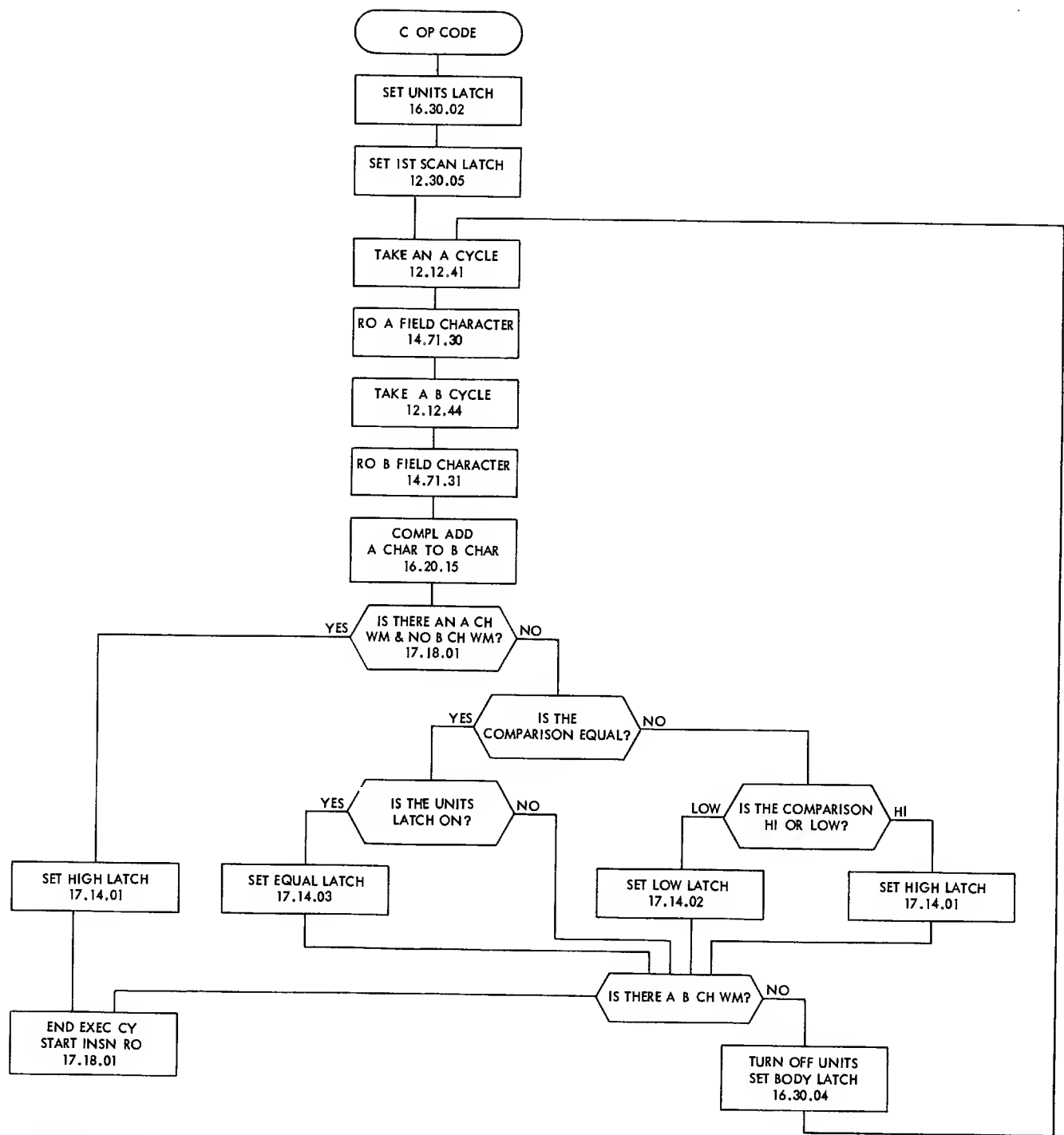
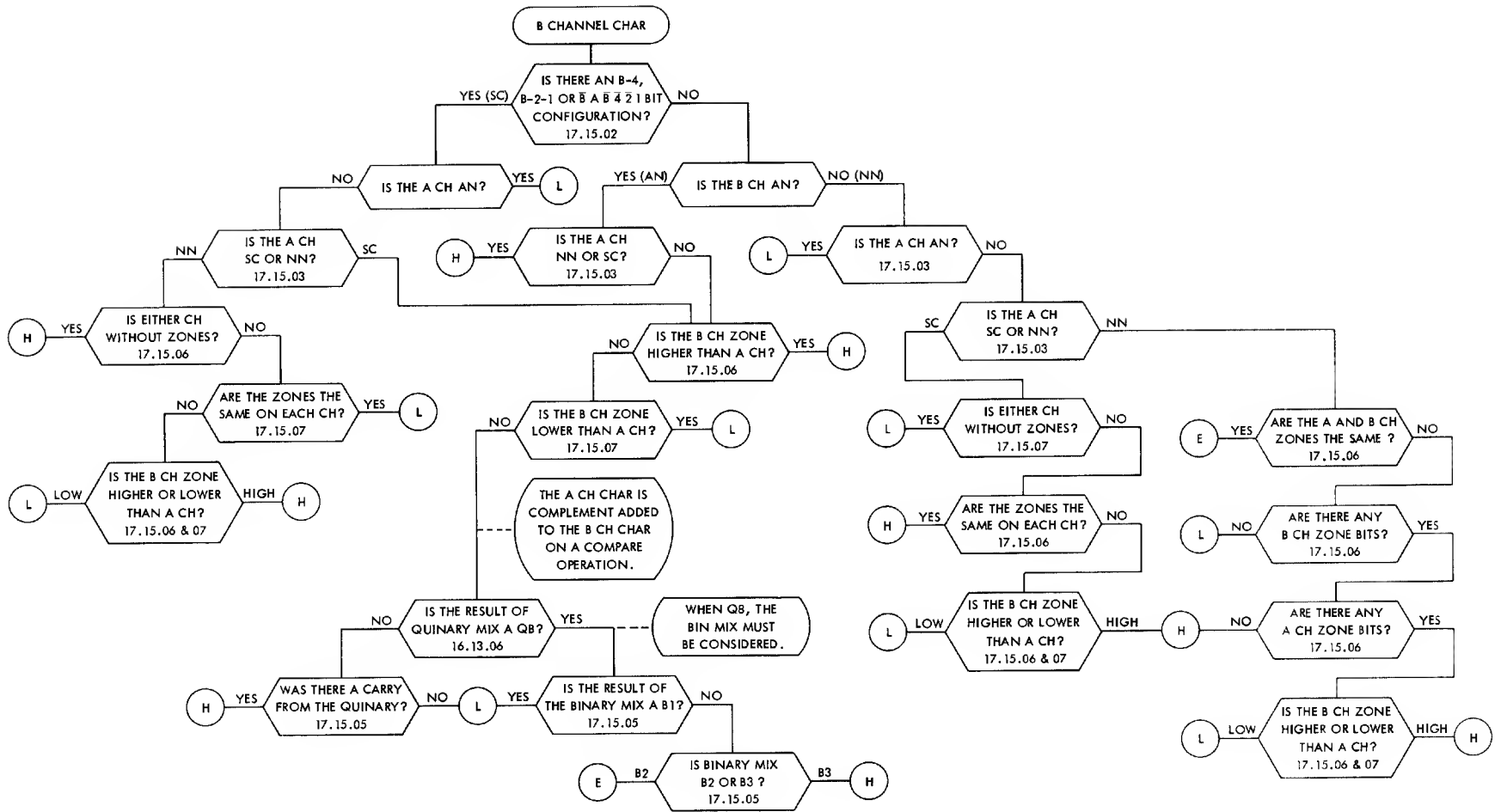


Figure 2.8-5 Compare



1. the blanking of zeros to the left of the first significant digit (Zero Suppress).
2. the insertion of asterisks in positions to the left of the first significant digit (Asterisk Fill).
3. the suppression of the decimal and of positions to the right when there are no significant digits (Decimal Control).
4. the placing of a dollar sign immediately to the left of the first significant digit (Floating Dollar).

The edit operation can require from one to three scans depending on the operation. Zero Suppression and Asterisk Fill require two scans. Decimal Control and Floating Dollar require three.

During the first scan, the characters in the control word determine:

1. where the characters from the A-field are placed.
2. whether Zero Suppress or Asterisk Fill is desired.
3. whether a floating dollar sign is desired.
4. whether to use a credit or a minus sign (depends on the sign of the A-field).

The operation starts by reading the units position of the A-field into the A-data register. The sign over the units position sets either the plus or minus latch. The first B-field character is read out and analyzed to determine what to do next. If the B-field character is anything other than a blank (b), zero (0), ampersand (&), C, R, or minus sign (-), the character is put right back into storage. The C, R, or minus sign is also stored, if the minus latch is ON; if the minus latch is OFF, a blank is stored. The ampersand means the space that also causes a blank to be stored. Any of these conditions cause another B-field character to be read out and analyzed.

Many B-field characters can be read out before a 0 or blank. The 0 or blank causes the A-field character to be stored. The units position of the A-field is stored without zones (sign). All of the control word up to the first zero or blank is called the status portion of the control word. A list of the special meanings of some characters in the status portion of the control word follows.

#### Status Portion of Control Word

<u>Character</u>	<u>Function</u>
b (blank)	Replaced with the character from the A-field.
0 (zero)	Replaced with the character from the A-field. Also indicates zero suppression. The rightmost zero in the control word indicates the rightmost limit of zero suppression.

C or R (credit)	If the sign of the A-field is plus, these two characters are replaced by blanks. When the sign is minus, they are stored back in place.
- (minus)	Same as CR.
& (ampersand)	Causes a blank space.

The portion of the control word from the first blank or zero to the position where the last A-field character is stored is called the body of the control word. In the body, certain characters take on different meanings as follows:

Body of Control Word

<u>Character</u>	<u>Function</u>
C or R (credit)	Stored back in regardless of sign.
- (minus)	Same as CR.
* (asterisk)	Replaced by an A-field character. If a zero preceded this *, an <u>Asterisk Fill</u> is called for.
& (dollar)	Replaced by an A-field character. If a zero preceded the \$, a <u>Floating Dollar</u> is called for.
b, 0, and &	The same as in the status portion.

The remainder of the control word is again considered the status portion. The functions of the characters in this status portion of the word are the same as in the first status portion except for the zero and blank that are stored back in, because there are no more A-field characters.

Example:

A-field	000109
	Status   Body   Status
Control word (B-field)	\$ b b , b b 0 . b b & C R & * *
Result of edit	\$ 1 . 0 9 C R * *



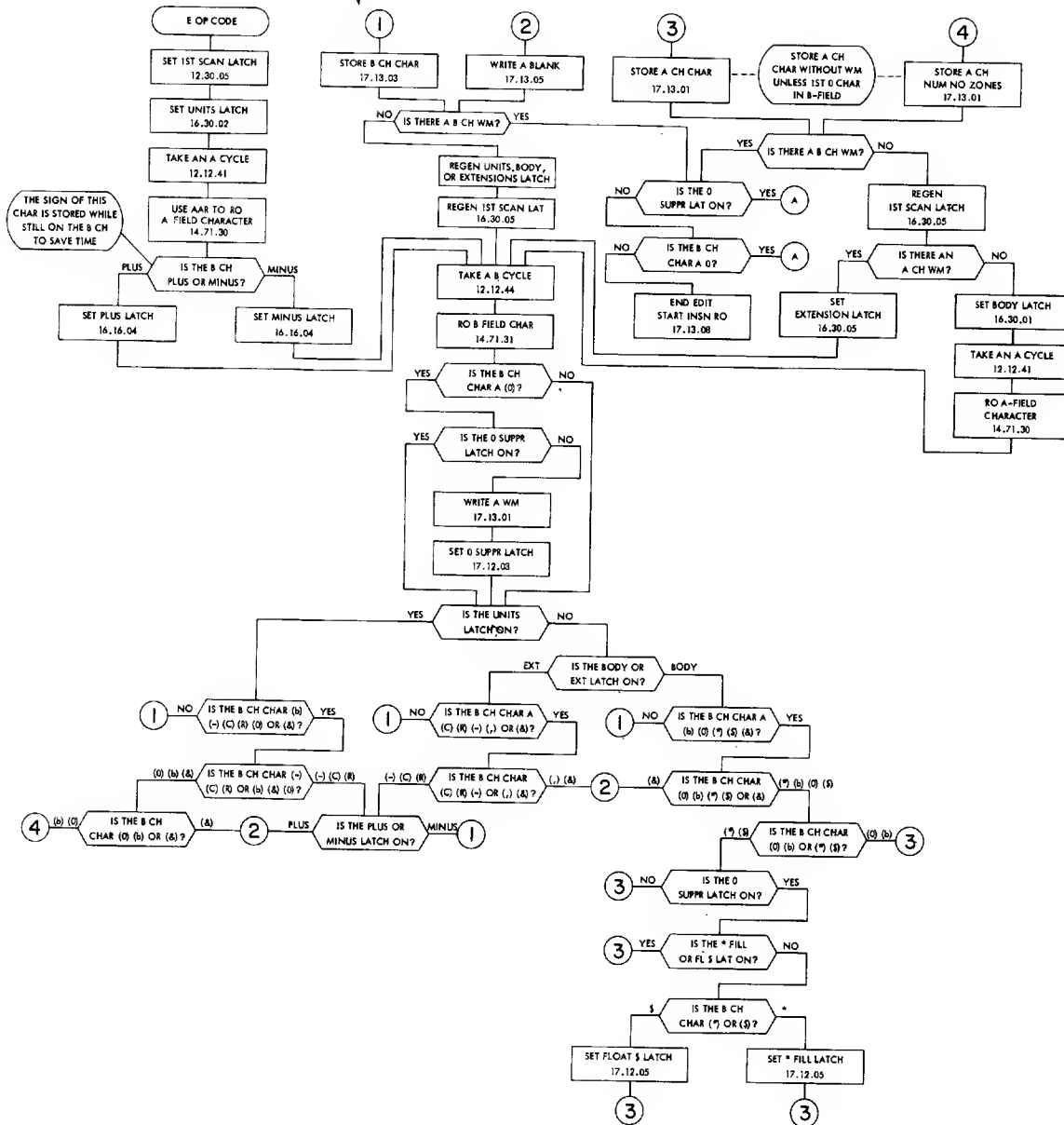


Figure 2.8-7 Edit 1st Scan

Common Op-code grouping lines are:

Instruction Read-out  
Not percent type  
Not addr dbl op codes  
2 addr no mod op codes  
Two address  
Addr type op codes  
No C or D-cycle op codes  
No D-cy at I-ring 6

Operational  
E or Z Op code  
No branch

Control  
1st scan first op codes  
A-cy first op codes  
Std A-cycle op codes  
A-reg to A-ch on B-cy ops  
Load mem on B-cy op codes  
Stop at J on B-cy op codes  
Read out BAR on scan B-cy ops  
Read out AAR on A-cy ops

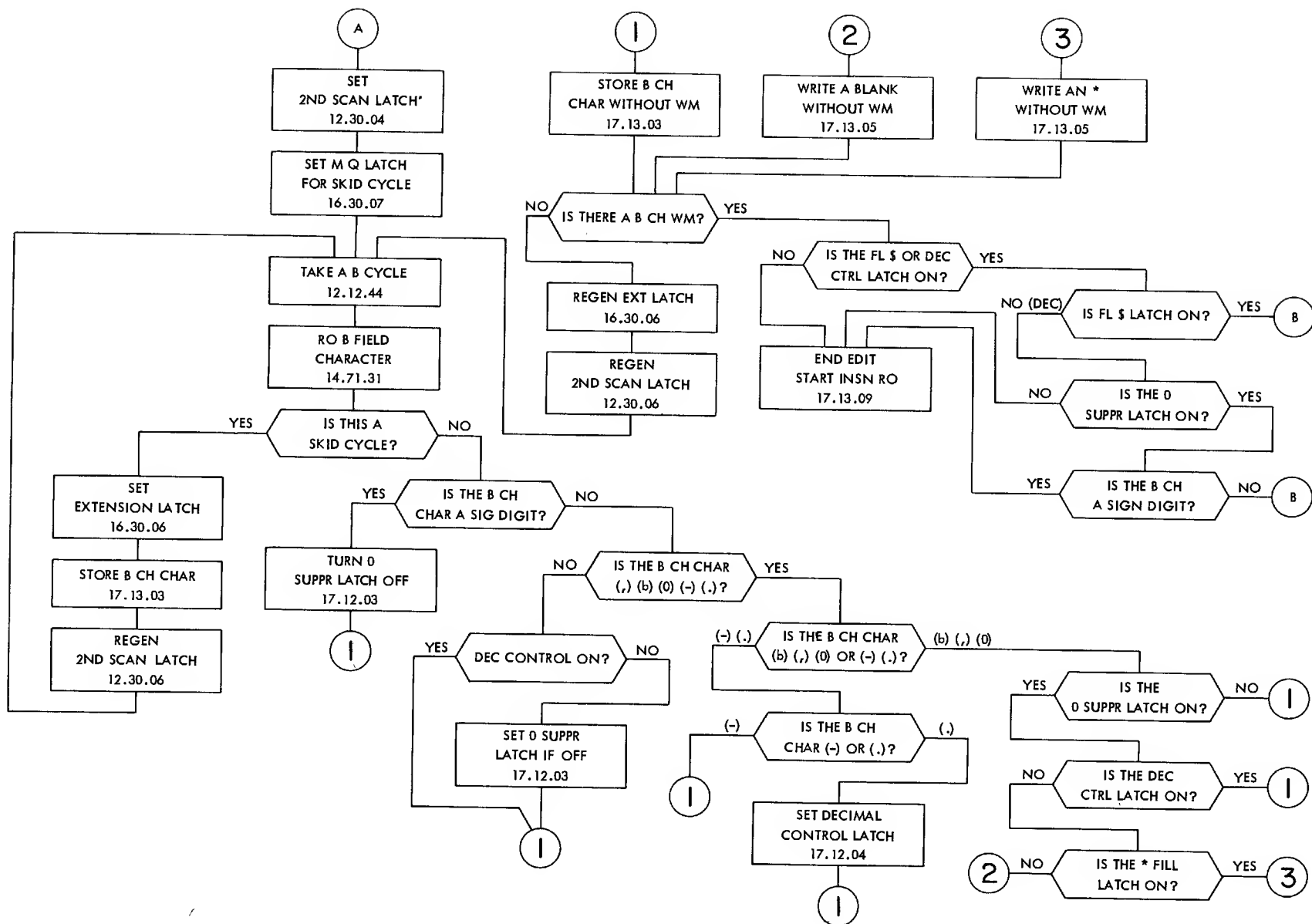
## Operation

The edit operation is started during Last Insn RO Cycle by setting the A-cycle, 1st scan, and Units Ctrl latches (Figure 2.8-7). The AAR is used to read out the A-field units-position character that is stored in the A-data register; and while still on the B-channel, it is analyzed to set the plus or minus latch. Because it is a Standard A-Cycle Ops, the 1st scan and units latches are regenerated and a B-cycle is initiated. The units latch signifies the status portion of the control word.

During this B-cycle the B-field character on the B-channel is analyzed to control the assembly unit to Store B-Ch Char, Write a Blank, or Store A-Ch Nu (No Zones). When the control-word character in the status portion controls the assembly to Store B-Ch Char or Write a Blank, the units latch is regenerated and another B-cycle is taken to read out the next position of the control word. When the character controls the assembly to Use A-Ch Nu, an A-cycle is initiated, and the body latch is set to identify the body of the control word.

Note on Figure 2.8-7, that the asterisk-fill or float-dollar-sign latches can only be set in the body and only if the zero-suppress latch is on. The zero-suppress latch is necessary to initiate the second scan. An A-channel WM causes the extension latch to come on after the last A-field character is stored. The rest of the control word is scanned until the B-Ch WM is found. This ends the first scan and the operation, if the zero-suppress latch is OFF. The zero-suppress latch ON starts the second scan (Figure 2.8-8) with a skid cycle, the same as in move zero suppress Op code, because the BAR address was modified to the address of the first character of the next field. The skid-cycle character is put back into storage, and another B-cycle is taken. If the B-channel character is a zero,

Figure 2.8-8 Edit 2nd Scan



blank, or comma, and if the zero suppress latch is on, a blank or asterisk (depending on the asterisk-fill latch) is stored. A decimal sets the decimal-control latch that prevents the blanking of any more zeros until the word mark that was written when the 0 was read on the first scan reads out. If any significant digits are read out after the decimal is read out, the zero-suppress latch is turned off, and the result is the decimal followed by the digits. For example, a one-cent field ends up .01 instead of . 1 because of decimal control. If there are no significant digits when the WM reads out, a 3rd scan is taken to blank out the zeros and decimal. The operation ends when the decimal is blanked.

There is another condition on zero suppress or asterisk fill. A character other than a significant digit or punctuation mark sets the zero-suppress latch back on. This allows zero suppression of multiple fields.

For example:                   0 0 1 0 0 C A R S \$ 0 2 0 , 0 0 0 . 0 0

Edit Results                   1 0 0 C A R S \$   2 0 , 0 0 0 . 0 0

The float-dollar latch also causes a 3rd scan (Figure 2.8-9).

Another skid cycle is taken, and the character is stored back into storage. On decimal control, the zero-suppress latch has to be on. The zeros are blanked or replaced by asterisks until the decimal is read out and blanked.

On float decimal, the characters are read out and stored back in, until a blank is sensed. The blank is replaced by a dollar sign, and the operation ends.

## 2.9.00 ARITHMETIC OP CODES

### 2.9.01 Add or Subtract (A or S)

#### Op-Code Function

The A-Op code causes the numerical data in the A-field to add algebraically to the numerical data in the B-field. The S-Op code causes the numerical data in the A-field to subtract algebraically from the numerical data in the B-field. The result is stored in true form in the B-field.

Op-code grouping lines:

<u>Instruction Read-out</u>
Not percent type
Addr double
Two addr no mod
Two addr
Addr type
No D-cy at I6

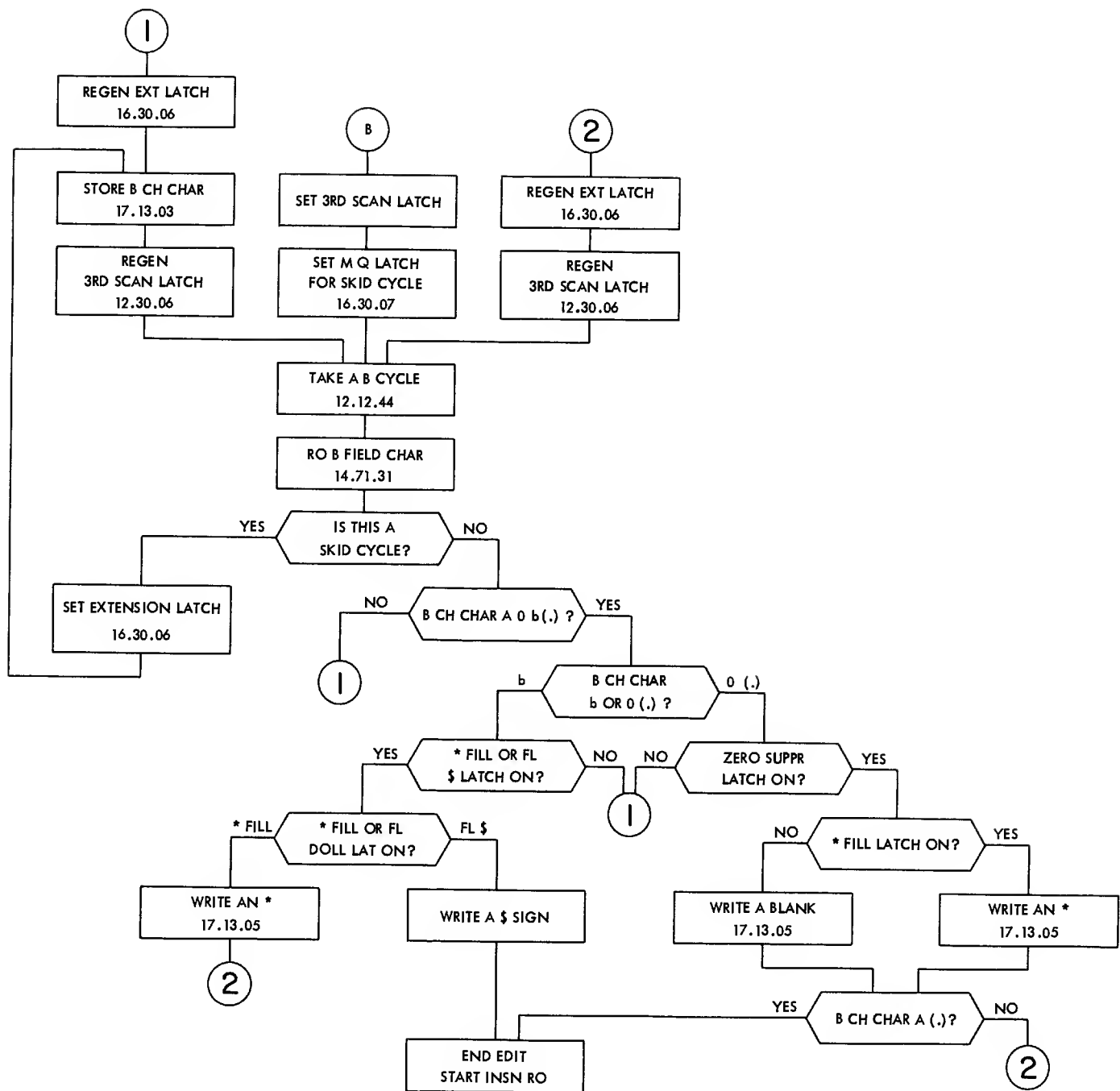


Figure 2.8-9 Edit 3rd Scan

<u>Operational</u>	
No branch	
Add or subtr op codes	
Arith type	
Add type	
<u>Control</u>	
Load mem on B-cy op codes	
1st scan first	
A-cy first	
Std A-cycle	
A-reg to A-ch on B-cy	
Read out AAR on A-cy	

#### Operation (Figure 2.9-1)

The operation starts with an A-cycle during which the units position of the A-field reads out of storage and is gated to the A-data register.

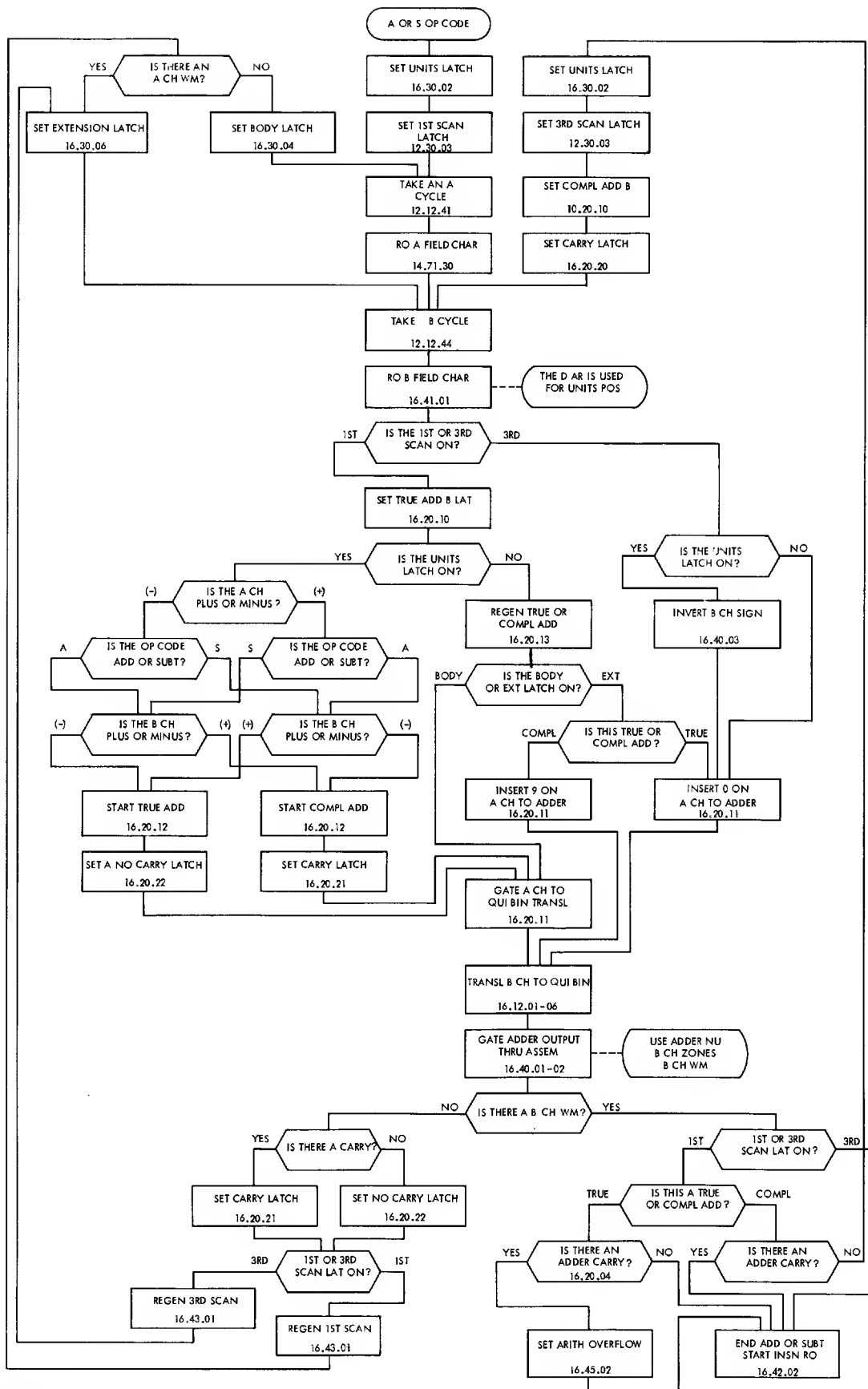
A B-cycle follows during which the units position of the B-field reads out onto the B-channel. The signs of the two channels and the type of Op code are analyzed to determine whether the A-field should be true or complement-added to the B-field (Figure 2.4-5). This first B-cycle is one logic gate longer to allow time to set up the true or complement controls. The two digits are combined in the adder, and the adder output is fed to the assembly. The assembly combines the numerical bits from the adder with the B-channel zones and word mark, if any, and routes the character to the B-field location in storage. If there is a carry, the carry latch is set and is combined with the binary portion of the next digits on the next B-cycle.

The operation continues to combine the two fields until an A-channel word mark is sensed. (A B-channel word mark stops the transfer if the A-field is longer than the B-field). The remainder of the B-field is combined with a zero on true-add or a nine-on-a-complement-add. The zero or nine is inserted directly into the adder on the A-field side.

When a B-channel word mark is sensed, the first scan and the operation are ended unless there is no carry from the high-order position on a complement-add. (A carry on a true-add signifies an overflow, and the internal indicator Arith Overflow is set.)

No-carry on a complement-add signifies that the sign of the B-field must be changed, and the field must be converted to the true form. For example:

Add	
A-field	- 17
B-field	+ 012
Complement-add A-field	982
	+
B-field	012
The carry latch is set ON	
for the units position	1
	+
	995



The B-field is in complement form with a plus sign. No-carry starts another forward scan (3rd scan latch). The B-field is complement-added to zeros inserted into the A-field side of the adder and the sign is inverted.

	+
B-field	995
Complement-add B-field	004
Insert zeros A-field	000
Carry latch ON for units position	1
	-----
	-
	005

Again the B-field word mark ends the scan and the operation.

## 2.9.02 Zero and Add or Zero and Subtract (? or !)

### Op-Code Function

The zero and add (ZA) Op code causes the numerical data in the A-field to transfer to the B-field. The result in the B-field has no zones except the A-field sign. The zero and subtract (ZS) Op code is the same as zero and add except that the resultant sign is the opposite of the A-field sign.

The sign over the units position is in the standard form; that is, a B-bit for minus or an A · B bit for plus.

Op-code grouping lines are:

#### Instruction Read-out

Not percent type  
Addr double  
Two addr no mod  
Two addr  
Addr type  
No D-cy at I6

#### Operational

No branch  
Reset type op codes  
Arith type  
Add type

#### Control

1st scan first  
A-cy first  
Std A-cycle  
A reg to A-ch on B-cy  
Read out AAR on A-cy  
Stop at J on B-cy op codes  
Load mem on B-cy op codes



## Operation (Figure 2.9-2)

The ZA or ZS operation starts with an A-cycle that causes the units position of the A-field to be set into the A-data register. During the following B-cycle, the units position of the B-field is read out, but a zero is inserted into the adder instead of the B-field character. The A-channel character is true-added to the B-field zero regardless of the signs of the field or of the Op code. The result from the adder is combined in the assembly with the sign of the A-field on ZA or with the opposite sign of the A-field on ZS. The operation continues: A-cycle, B-cycle, A-cycle, etc. using no zones and using the adder output. An A-channel word mark ends the A-cycles, and the remainder of the B-field is filled with zeros until a B-channel word mark is sensed. This ends the operation.

### 2.9.03 Multiply (@)

#### Op-Code Function

The multiply Op code causes the numerical data in the A-field to repetitively add in the B-field. The multiplier is located in the high-order positions of the B-field. The low-order positions of the B-field are used to develop the product. The number of positions used for the product must be one higher than the number of positions in the A-field (multiplicand).

To multiply  $82 \times 250$  (A-field) the B-field must be  $820000^v$ . The A- and B-field addresses locate the units position of their respective fields. The result of the multiplication (product), is located in the B-field with the sign over the units position.

#### Theory of Multiplication

Multiplication in the IBM 1410 is a series of repetitive true or complement-additions. The number and type of additions are determined by the specific multiplier digits.

Digits 1-4. A multiplier units digit of from one through four causes a corresponding number of true additions of the A-field (multiplicand). These true-add scans start in the units position of the product field. Before each add scan, the multiplier digit is reduced by one until the digit is zero. When this occurs, a left shift is forced. The left shift causes:

1. The tens position of the multiplier (any digit one through four) to control the number of true-add scans.
2. The true-add scans to start in the tens position of the product field.

This procedure is repeated for each position of the multiplier, until the multiplier is all zeros. At the end of the operation the B-field contains only the completed product and the multiplier has been eliminated.

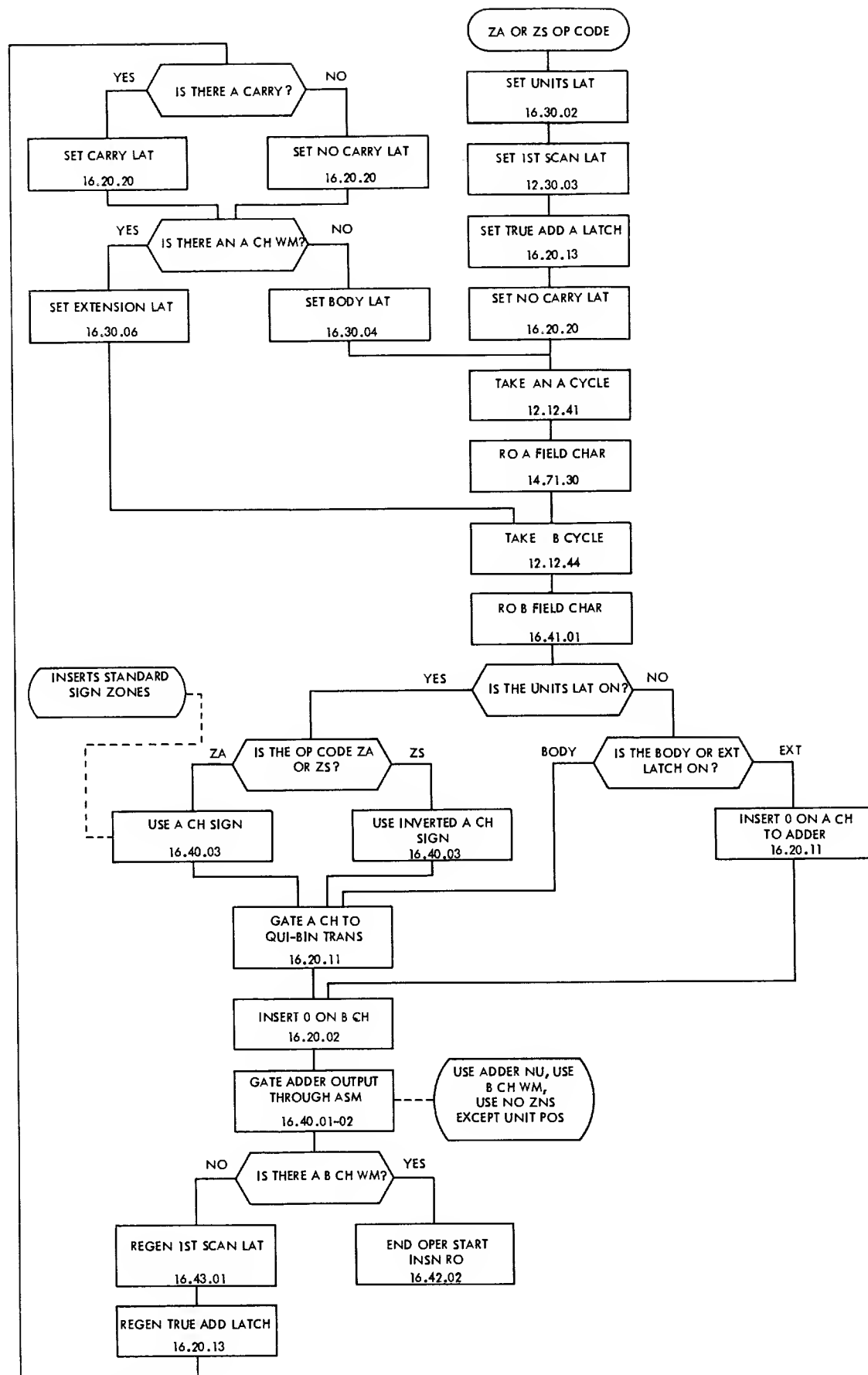


Figure 2.9-2 Zero and Add, Zero and Subtract

Example:  $203 \times 1625$

	multiplier	B-field
	20300000	
Read out 3 and reduce	20200000	
True-add	01625	
Read out 2 and reduce	20101625	
True-add	01625	
Read out 1 and reduce	20003250	
True-add	01625	
Read out 0 and shift	20004875	
Read out 0 and shift	20004875	
Read out 2 and reduce	10004875	
True-add	01625	
Read out 1 and reduce	00167375	
True-add	01625	
Read out 0, <u>End Oper</u>	00329875	

Digits 5-9. Any multiplier digit of five or more causes the multiplicand to be complement-add scanned in developing the product. The number of complement scans is determined by the specific multiplier digit. These complement-add scans are followed by a left shift of the multiplicand in the product and a true-add scan.

For example, when multiplying by seven, the multiplicand is complement-add scanned three times. Each of these scans starts in the units position of the product. The multiplicand is then left-shifted one position in the product and true-add scanned once. This true-add scan starts in the tens position of the product. This effectively multiplies the number by ten minus three, which is seven.

Example:  $7 \times 1625$

1625	1625
$\times 7$	$\times 10$
<u>11375</u>	16250
	-1625
	<u>14625</u>
	-1625
	<u>13000</u>
	-1625
	<u>11375</u>

This method for processing of multiplier digits five through nine permits considerable saving of processing cycles and time. The example described in Digits 1-4 requires seven true scans compared with one true scan and three complement scans actually used.

TA	1625	TA	16250
TA	<u>1625</u>	CA	<u>98375</u>
	3250		14625
TA	<u>1625</u>	CA	<u>98375</u>
	4875		13000
TA	<u>1625</u>	CA	<u>98375</u>
	6500		11375
TA	<u>1625</u>	Actual machine operation	
	8125		
TA	<u>1625</u>		
	9750		
TA	<u>1625</u>		
	11375		
True-add scans			

For a comprehensive picture of the steps involved in processing any multiplier, consider the multiplier 1991. In addition to the normal-add scans (true or complement) indicated by the multiplier digits, several additional add scans are required. These additional add scans occur following a left shift, when the next multiplier digit indicates a change from true to complement or the reverse. In this multiplier an additional complement-add scan is required in the tens position, and an additional true-add scan is required in the thousands position. When an additional add-scan occurs, the multiplier digit is neither increased nor decreased during that scan.

Processing the multiplier in this manner actually employs (in the multiply operation) a multiplier of 2000 minus 10 plus 1, which is 1991.

$$\begin{array}{r}
 2000 \\
 - 10 \\
 \hline
 1990 \\
 + 1 \\
 \hline
 1991
 \end{array}$$

The following steps are taken by the CPU in developing this multiplier.

1. The units position 1 requires one true-add scan and a reduction to zero. The units zero signals a left shift.
2. The tens position 9 signals a change from true to complement. This change requires one complement-add scan with no multiplier digit increase. (Because 9 is the complement of zero, no further increase in the 9-multiplier digit occur). Following this scan, the 9 signals a left shift.
3. The hundreds position 9, while in the complement-add scan, immediately signals a left shift.

4. The thousands position 1 signals a change from complement to true. This change requires one true-add scan with no multiplier digit reduction. This scan is followed by a second true-add scan and a reduction of the thousands position 1 to zero. This zero, with a B-field word mark, signals the end of the multiply operation.

Multiplier example: 1 9 9 1

1	TA scan ·	-1
9 0 shift ·	CA scan ·	$\overline{1}$
9 9 0 shift ·	$\overline{CA}$ scan ·	$\overline{1}$
1 9 9 0 shift ·	TA scan ·	$\overline{-1}$
0 0 0 0	TA scan ·	End oper

The fact that all multiply operations start and end in true-add is helpful in determining when extra add scans are required. This rule applies even when the units and/or highest-order positions of the multiplier are in the five-through-nine-digit range.

Example:  $98 \times 1625$

	9800000
Read out 8 and set <u>Compl Add</u>	9800000
<u>Compl Add</u>	98375
Read out 8 and increase	9998375
<u>Compl Add</u>	98375
Read out 9 and shift	9996750
Read out 9 and shift set <u>True Add</u>	9996750
<u>True Add</u>	01625
End operation	0159250

Common Op-code grouping lines are:

Instruction Read-out  
 Not percent type op codes  
 Not addr dbl op codes  
 2 addr no mod op codes  
 2 address op codes  
 Addr type op codes  
 C-cycle op codes

Operational  
 Mpy or div op codes  
 Arith type op codes  
 No-branch op codes

#### Control

1st scan first op codes  
A-cycle first op codes  
Std A-cycle op codes  
A reg to A-ch on B-cy ops  
Load mem on B-cycle op codes  
Stop at J on B-cycle op codes

#### Operation (Figures 2.9-3, 2.9-4)

The first objective is to find and to read out the units position of the multiplier. This is located immediately to the left of the product field. To find the multiplier, the CPU scans through the A- and B-fields (sets the product field to zero) until the A-field word mark is sensed. The A-channel word mark causes the extension latch to be set, and a C- and B-cycle to be taken. During the C-cycle, the CAR is used to read out the units position of the A-field. This character is stored in the A-data register. The B-cycle reads out, and sets the extra position of the product field to zero. Another B-cycle follows, and the MQ latch sets to identify this B-field character as the units position of the multiplier. During this B-cycle, the signs over the units positions of the multiplier and multiplicand (these were stored in the A-data register during the previous A-cycle) are analyzed to determine and to set the product sign into the sign latches.

Also during this B-cycle the multiplier digit is analyzed to determine whether a true- or complement-add is to be taken. If the multiplier is from five through nine, the B-channel character (multiplier) is stored unchanged and the complement latch is set. A multiplier digit from one through four sets the true latch again, and causes the multiplier digit to be decreased by one. This is accomplished by inserting a nine into the A-channel side of the adder. This is added to the B-channel, and effectively subtracts one from the multiplier.

The multiplier digit also determines whether the no-scan latch, 1st scan latch, or 2nd scan latch should be ON during the next cycle that is a D-cycle. During this D-cycle, the DAR is used to read out the units positions of the B-field. The sign latch that was set during the last B-cycle is used to set the sign over the units position of the product.

The 2nd scan latch is turned on when a zero multiplier with a word mark is sensed. The multiply operation ends with the completion of this D-cycle.

A zero multiplier without a word mark regenerates the 1st scan latch that modifies the address in the DAR by minus one (shift one position to the left), and initiates another B-cycle with the MQ latch ON. During this cycle the BAR reads out the next character of the multiplier.

If the multiplier was a one-through-nine, the no-scan latch is set. The no-scan modifies the DAR address by zero (no shift) and sets the units and 3rd scan latches. The operation continues by scanning the A- and B-fields to true- or complement-add the A-field to the B-field.

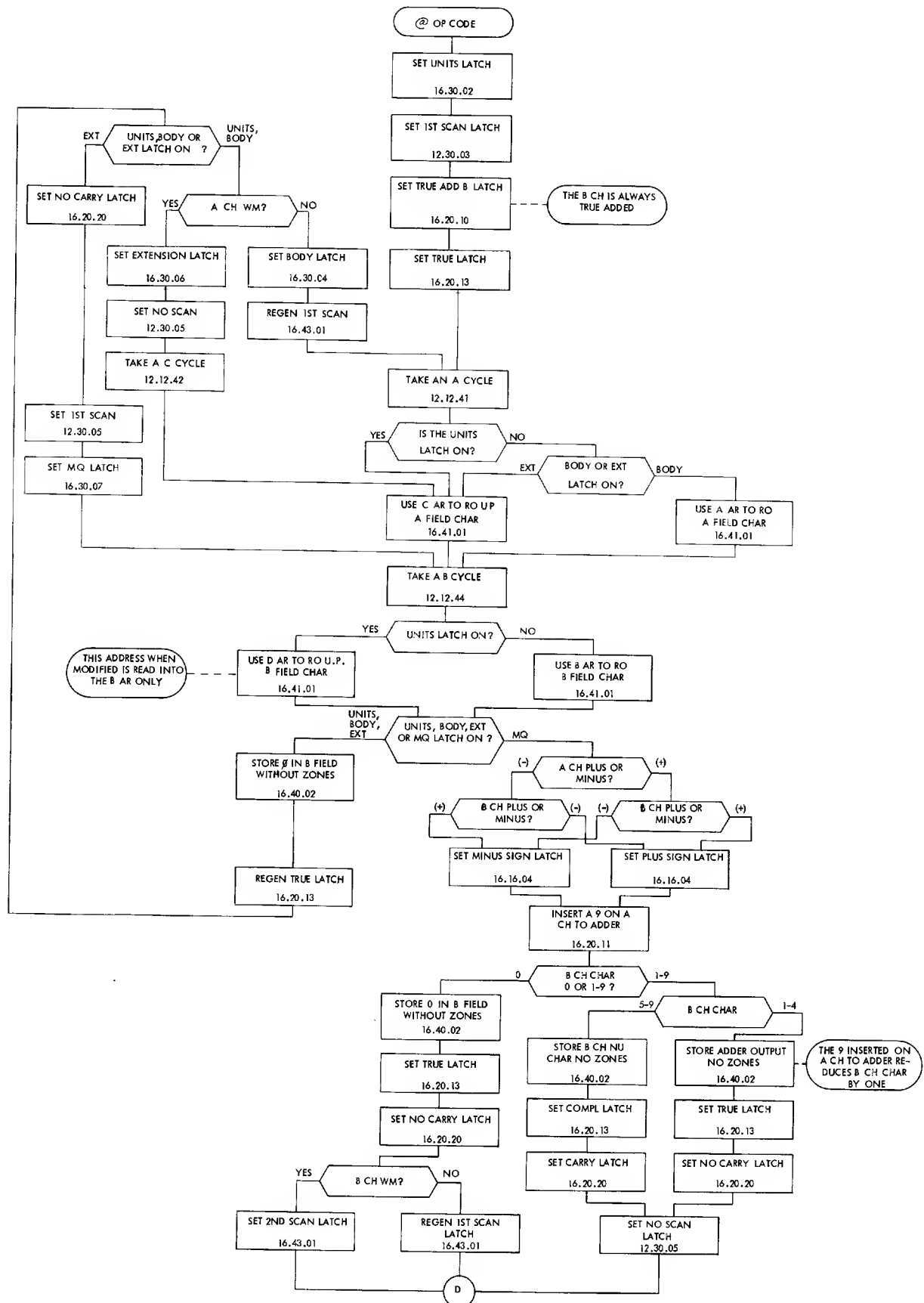


Figure 2.9-3A Multiply 1st Scan

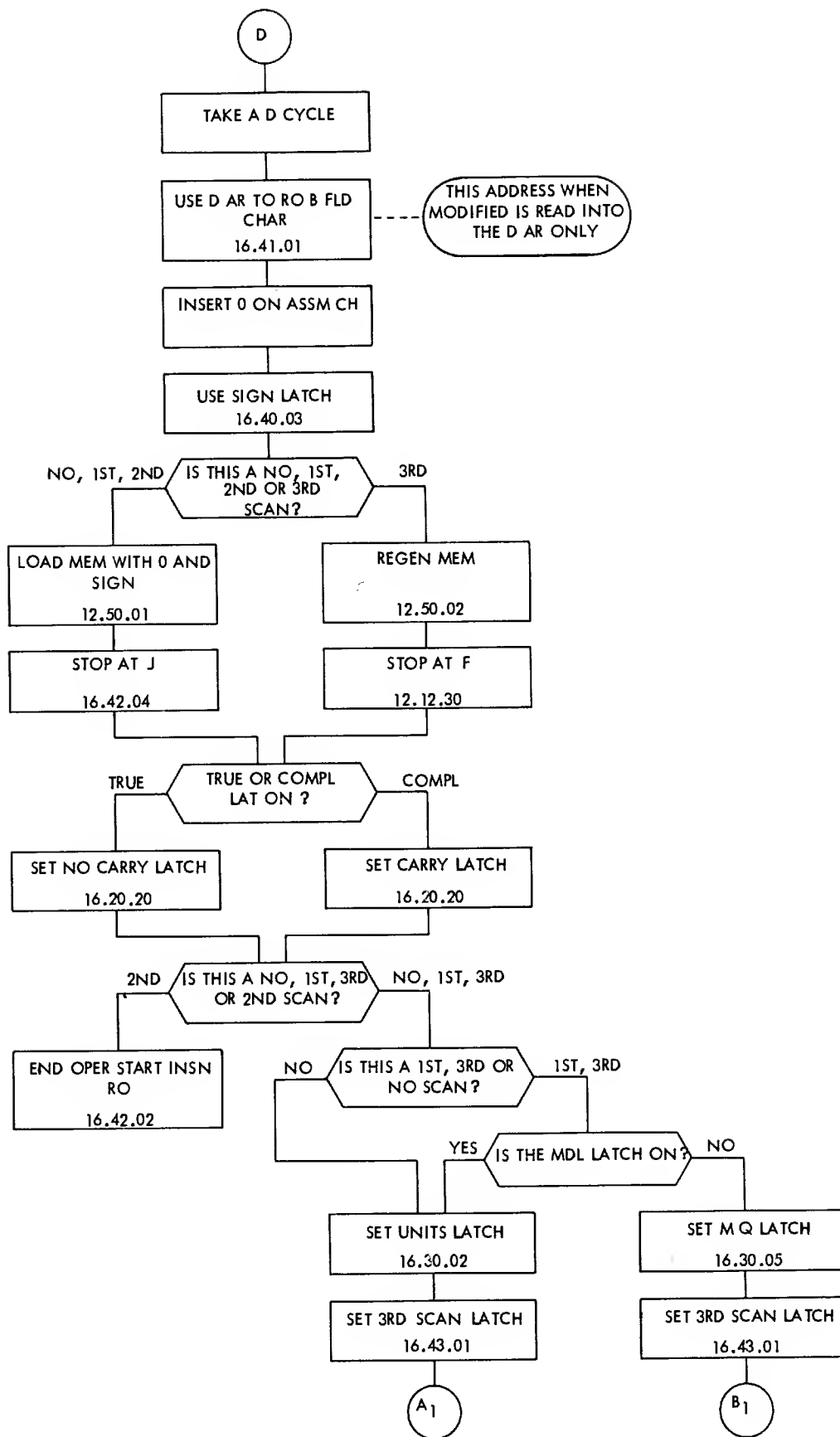


Figure 2.9-3B Set Sign and/or Shift



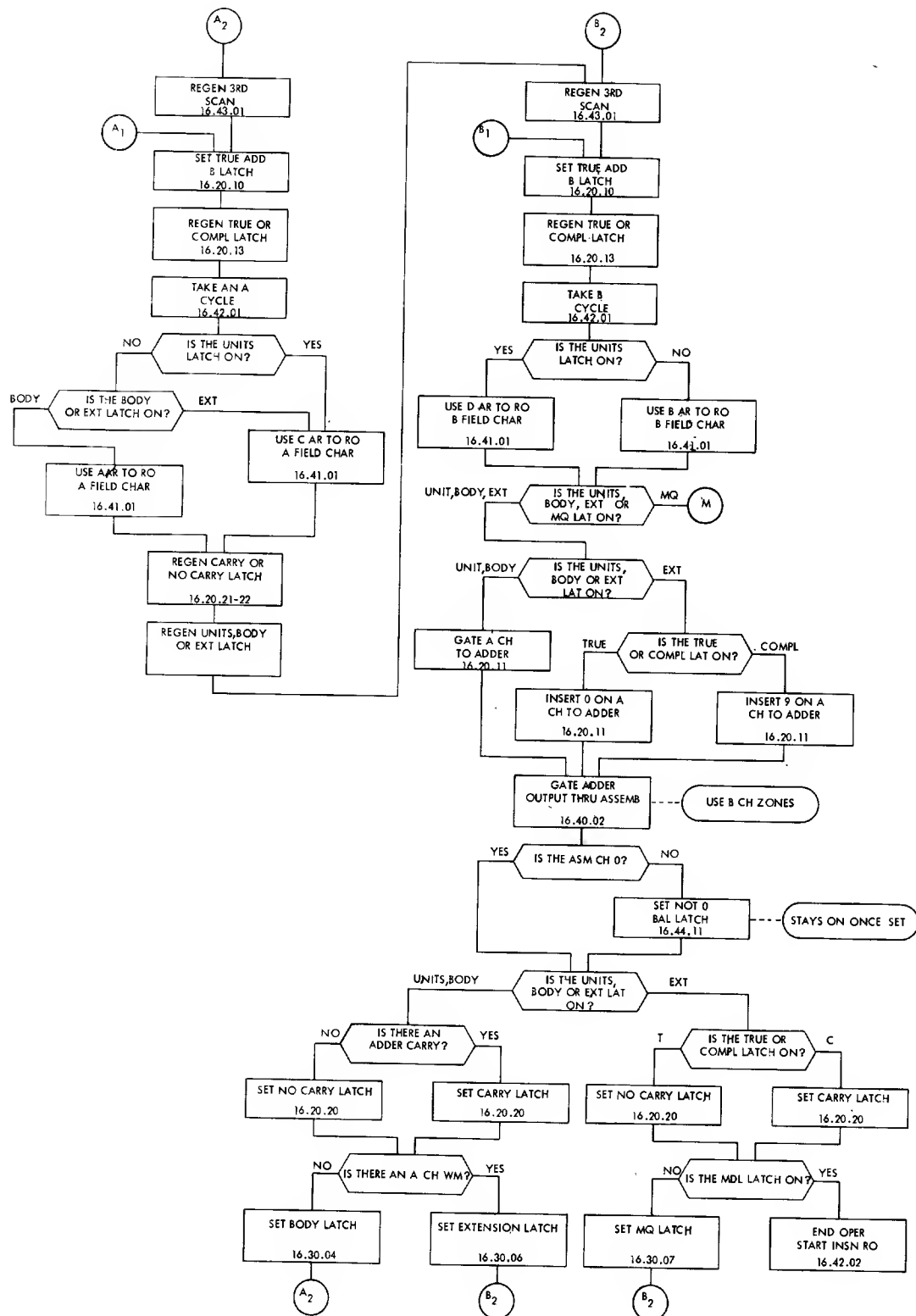
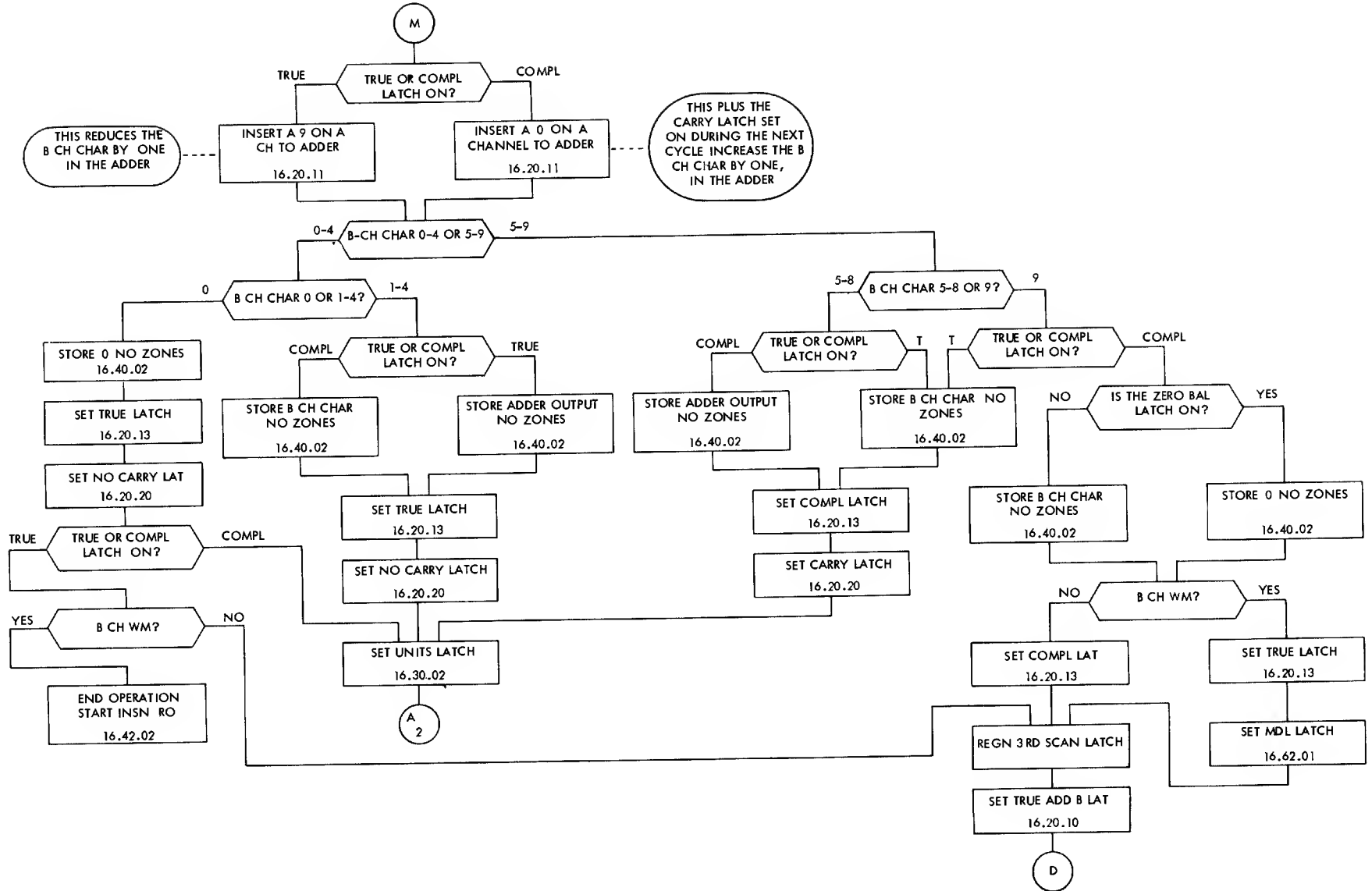


Figure 2.9-3C Add A-Field to B-Field

Figure 2.9-3D MQ Controls



MULTIPLY PROBLEM:

"B" FIELD

V B 2

"A" FIELD

V 1 8 1

▲ -LOCATION OF D AR

OBJECTIVES	CYCLE	A/R RO	UNITS BODY EXTN MQ	SCAN 1ST 2ND, 3RD NO	STORED "B" FIELD	DIGIT ON ASSEM	ADDER "B"	CARRY NO CARRY	ADDER "A"	TRUE OR COMP	REMARKS
LOCATE- UNITS POS. OF MULTIPLIER	A B	C AR D AR	U	1	V +	0	0			T	UNITS A
ZERO- PRODUCT FIELD	A B	A AR B AR	Y	1		0	0			T	UNITS B
	A B	A AR B AR	Y	1		0	0			T	
	A B	C AR B AR	X	1		0	0			T	UNITS A
ANALYZE-SIGN & MULTIPLIER	B	B AR	MQ	1	1		1	2	C	9	REDUCE
STORE-SIGN	D	D AR		N		0	0			T	UNITS B
					V B 1 0 0 0 0						
TRANSFER- TRUE ADD SCAN	A B	C AR D AR	U	3		1	1	0	C	1	UNITS A
	A B	A AR B AR	Y	3		8	B	0	C	B	UNITS B
	A B	A AR B AR	Y	3		1	1	0	C	1	
	B	B AR	X	3		0	0	0	C	0	
	B	B AR	MQ	3		0	0	1	C	9	REDUCE
					V B 0 0 1 B 1						
TRANSFER- TRUE ADD SCAN	A B	C AR D AR	U	3		2	2	1	C	1	UNITS A
	A B	A AR B AR	Y	3		6	6	8	C	8	UNITS B
	A B	A AR B AR	Y	3		3	3	1	C	1	
	B	B AR	X	3		0	0	0	C	0	
	B	B AR	MQ	3		0	0	0	C	9	
MODIFY D AR -1	D	D AR		3			0			T	SHIFT
ANALYZE-MULT.CHAR.	B	B AR	MQ	3	B		8	8	C	9	
					V B 0 0 3 6 2						
TRANSFER- COMP ADD SCAN	A B	C AR D AR	U	3		5	5	6	C	8	UNITS A
	A B	A AR B AR	Y	3		5	5	3	C	1	TENS B
	A B	A AR B AR	Y	3		8	8	0	C	B	
	B	B AR	X	3		9	9	0	C	9	
	B	B AR	MQ	3		9	9	B	C	0	INCREASE
					V B 9 8 5 5 2						
TRANSFER- COMP ADD SCAN	A B	C AR D AR	U	3		4	4	5	C	B	UNITS A
	A B	A AR B AR	Y	3		7	7	5	C	1	TENS B
	A B	A AR B AR	Y	3		6	6	8	C	B	
	B	B AR	X	3		9	9	9	C	9	
	B	B AR	MQ	3		9	9	9	C	0	
MODIFY D AR -1	D	D AR		3						T	SHIFT & MDL - "ON"
					V B 9 6 7 4 2						
TRANSFER- TRUE ADD SCAN	A B	C AR D AR	U	3		B	8	7	C	1	UNITS A
	A B	A AR B AR	Y	3		4	4	6	C	8	HUNDREDS B
	A B	A AR B AR	Y	3		1	1	9	C	1	
END OPERATION	B	B AR	X	3		0	0	9	C	0	
					V B 1 4 B 4 2						

Figure 2.9-4 Multiply Chart

#### 2.9.04 Divide (%)

##### Op-Code Function

The divide Op code causes the dividend located in the B-field to be divided by the divisor in the A-field. The result or quotient is located in the high-order positions of the B-field with any remainder still in the low-order of the B-field. The length of the dividend-quotient field (B-field) is one more than the total number of positions in the divisor and dividend fields. The B-address locates the high-order position of the dividend in the B-field. The A-address locates the units position of the A-field. The quotient should be preset to zeros. The dividend must have a standard sign; that is, an A- and B-bit for a plus, or a B-bit only for a minus field.

##### Theory of Divide

The development of the quotient involves counting the number of repetitive subtractions (reductions) required to reduce the dividend to less than the divisor. Any portion of the dividend that exists in the low-order B-field positions at the conclusion of the operation is the remainder.

The divide operation is initiated by a reduction attempt in the high-order position of the dividend. If this reduction is successful, it is followed by another reduction attempt in the same dividend position. Any unsuccessful reduction (divisor larger than dividend) is followed by a true-add scan (correction) to restore the previous dividend figure. The unsuccessful reduction is called an overdraw.

A correction cycle is normally followed by a right shift to allow processing of the next lower-order position of the dividend. This is true for all but the units position of the dividend where the presence of a B-bit signals the end of the dividend.

The accumulation of the quotient (in the high-order of the B-field) occurs at the end of each successful reduction. This accumulation continues in the same quotient position, until a right shift of the divisor in the dividend field. At this time, a similar shift occurs in the quotient and further accumulation now occurs in this new position.

The following example covers the various cycles involved in a divide operation:

Example: 14980 65

		v
Reduction	CA	0 0 0 1 4 9 8 0
		9 3 5
Overdraw		0 9 3 6 4 9 8 0
Correction	TA	0 6 5
		0 0 0 1 4 9 8 0
Shift and reduction	CA	9 3 5
Overdraw		0 0 9 4 9 9 8 0
Correction	TA	0 6 5
		0 0 0 1 4 9 8 0
Shift and reduction	CA	9 3 5
Accumulate quot		0 0 1 0 8 4 8 0
Reduction	CA	9 3 5
Accumulate quot		0 0 2 0 1 9 8 0
Reduction	CA	9 3 5
Overdraw		0 0 2 9 5 4 8 0
Correction	TA	0 6 5
		0 0 2 0 1 9 8 0
Shift and reduction	CA	9 3 5
Accumulate quot		0 0 2 1 1 3 3 0
Reduction	CA	9 3 5
Accumulate quot		0 0 2 2 0 6 8 0
Reduction	CA	9 3 5
Accumulate quot		0 0 2 3 0 0 3 0
Reduction	CA	9 3 5
Overdraw		0 0 2 3 9 3 8 0
Correction	TA	0 6 5
		0 0 2 3 0 0 3 0
Shift and reduction	CA	9 3 5
Overdraw		0 0 2 3 0 9 6 5
Correction	TA	0 6 5
		v - +
End divide		0 0 2 3 0 0 3 0
Quotient		Remainder

Common Op-code grouping lines are:

Instruction Read-out  
Not percent type op codes  
Not addr dbl op codes  
2 addr no mod op codes  
2 address op codes  
Addr type op codes  
C-cycle op codes

Operational  
Mpy or div op codes  
Arith type op codes  
No-branch op codes

Control  
1st scan first op codes  
A-cycle first op codes  
Std A-cycle op codes  
A-reg to A-ch on B-cy ops  
Load mem on B-cy op codes  
Stop at J on B-cy op codes

Operation (Figures 2.9-5, 2.9-6)

The DAR provides the address of the high-order dividend position for the initial reduction attempt. When a right shift occurs the DAR is modified by 1 to permit subsequent reductions to occur in the next lower-order position of the dividend. The CAR is always used to locate the units position of the A-field. Reductions continue until an overdraw occurs. An overdraw is recognized by the lack of a carry when in a reduction (CA scan) with the extension latch on.

An overdraw is always followed by a correction (TA scan). This restores the dividend to a true figure, and is followed by a right shift (except when a B-bit is sensed in the dividend).

Each successful reduction is followed by an increase of one in the quotient accumulation. This is done by combining the carry (from the successful reduction) with an A-channel zero and the B-channel numerical bits (previous quotient digit). The MQ latch is on when this quotient increase occurs. To provide for accumulation of all the digits in the quotient, each right shift in the dividend causes a similar shift in the quotient.

When a B-bit is sensed in the dividend during a correction (TA scan) the MDL (multiply-divide last) latch is set, and the proper quotient sign is set in the sign latches. Following the correction, one additional B-cycle is taken with the MQ latch ON. This stores the quotient sign (previously set in the sign latches) over the units position of the quotient. The divide operation ends at this point.







REDUCTION- COMP ADD SCAN	A B	C AR D AR	U	3			5	5	0	C	4	S	UNITS A TENS B
	A B	A AR B AR	Y	3			2	2	1	C	1	S	
	B	B AR	X	3		1		1	2	C	9	S	
ACCUMULATE QUOT	B	B AR	MQ	3		7		7	6	C	0	S	
0 0 7 1 2 5 9													
REDUCTION- COMP ADD SCAN	A B	C AR D AR	U	3			0	0	5	C	4	S	UNITS A TENS B
	A B	A AR B AR	Y	3			4	4	2	C	1	S	
	B	B AR	X	3		0		0	1	C	9	S	
ACCUMULATE QUOT	B	B AR	MQ	3		8		8	7	C	0	S	
0 0 8 0 4 0 9													
REDUCTION- COMP ADD SCAN	A B	C AR D AR	U	3			5	5	0	C	4	S	UNITS A TENS B
	A B	A AR B AR	Y	3			5	5	4	C	1	S	
	B	B AR	X	3		9		9	0	C	9	S	
OVERDRAW	B	B AR	X	3		9		9	0	C	9	S	
0 0 8 9 5 5 9													
CORRECTION-TRUE ADD SCAN	A B	C AR D AR	U	3			0	0	5	C	5	T	UNITS A TENS B
	A B	A AR B AR	Y	3			4	4	5	C	8	T	
	B	B AR	X	3		0		0	9	C	0	T	
0 0 8 0 4 0 9													
MODIFY D AR + 1	D	D AR		2			1						SHIFT
REDUCTION- COMP ADD SCAN	A B	C AR D AR	U	3			4	4	9	C	4	S	UNITS A UNITS B
	A B	A AR B AR	Y	3			2	2	0	C	1	S	
	B	B AR	X	3		3		3	4	C	9	S	
ACCUMULATE QUOT	B	B AR	MQ	3		1		1	0	C	0	S	
0 0 8 1 3 2 4													
REDUCTION- COMP ADD SCAN	A B	C AR D AR	U	3			9	9	4	C	4	S	UNITS A UNITS B
	A B	A AR B AR	Y	3			3	3	2	C	1	S	
	B	B AR	X	3		2		2	3	C	9	S	
ACCUMULATE QUOT	B	B AR	MQ	3		2		2	1	C	0	S	
0 0 8 2 2 3 9													
REDUCTION- COMP ADD SCAN	A B	C AR D AR	U	3			4	4	9	C	4	S	UNITS A UNITS B
	A B	A AR B AR	Y	3			5	5	3	C	1	S	
	B	B AR	X	3		1		1	2	C	9	S	
ACCUMULATE QUOT	B	B AR	MQ	3		3		3	2	C	0	S	
0 0 8 3 1 5 4													
REDUCTION- COMP ADD SCAN	A B	C AR D AR	U	3			9	9	4	C	4	S	UNITS A UNITS B
	A B	A AR B AR	Y	3			6	6	5	C	1	S	
	B	B AR	X	3		0		0	1	C	9	S	
ACCUMULATE QUOT	B	B AR	MQ	3		4		4	3	C	0	S	
0 0 8 4 0 6 9													
REDUCTION- COMP ADD SCAN	A B	C AR D AR	U	3			4	4	9	C	4	S	UNITS A UNITS B
	A B	A AR B AR	Y	3			8	8	6	C	1	S	
	B	B AR	X	3		9		9	0	C	9	S	
OVERDRAW	B	B AR	X	3		9		9	0	C	9	S	
0 0 8 4 9 8 4													
CORRECTION-TRUE ADD SCAN	A B	C AR D AR	U	3			9	9	4	C	5	T	UNITS A UNITS B
	A B	A AR B AR	Y	3			6	6	8	C	8	T	
	B	B AR	X	3		0		0	9	C	0	T	
END OPERATION	B	B AR	MQ	3		4		4	4	C	9	T	
0 0 8 4 0 6 9													
QUOTIENT REMAINDER													

Figure 2.9-6B Divide Chart

## 2.10.00 LOGIC OP CODES

The IBM 1410 can be programmed to test for conditions that may arise during processing and to transfer (branch) the program to an alternate set of instructions (subroutine). The Op code and d-modifier determine which condition is tested. Some of the conditions that can be tested by programming are:

1. Internal conditions (such as zero balance, compare latches, divide overflow, etc.)
2. I/O channel indicators (such as channel busy, not ready, data check, etc.)
3. Word marks.

A transfer from one instruction to another instruction or set of instructions to alter the execution in sequence of program steps is called a program branch.

### 2.10.01 Branch Operation (Figure 2.10-1)

A branch instruction has an I-address in place of the A-field address. During instruction read-out this I-address is stored in the AAR. The I-address is the location of the Op code of the instruction to which the program branches, if the conditions tested call for a branch.

When a branch is required, the CPU takes a B-cycle. During this B-cycle the IAR reads out to the STAR and through the modifier to the BAR. A modify-by-zero condition exists in the modifier because the no-scan latch is set.

Note: The new contents of the BAR, which is the address of the next sequential instruction (NSI), can be stored in cores by writing the first instruction of the subroutine as a G-Op code. In this case, the last instruction in the subroutine is an unconditional branch:

J (I) BLANK to the stored NSI address.

The branch to AAR latch is set to cause the AAR to read out to STAR for the I-Op cycle of instruction read-out. This is the start of normal instruction read-out for the first branch routine instruction.

### 2.10.02 Branch if Internal Indicator On (J)

#### Op-Code Function

The J-Op code allows the program to test the internal indicators. The d-character modifier specifies the condition to be tested. If the indicator tested is OFF, the program continues on to the NSI. If the indicator is ON the program branches to the I-address.

Figure 2.10-2 shows the d-modifiers and the indicators that they test.

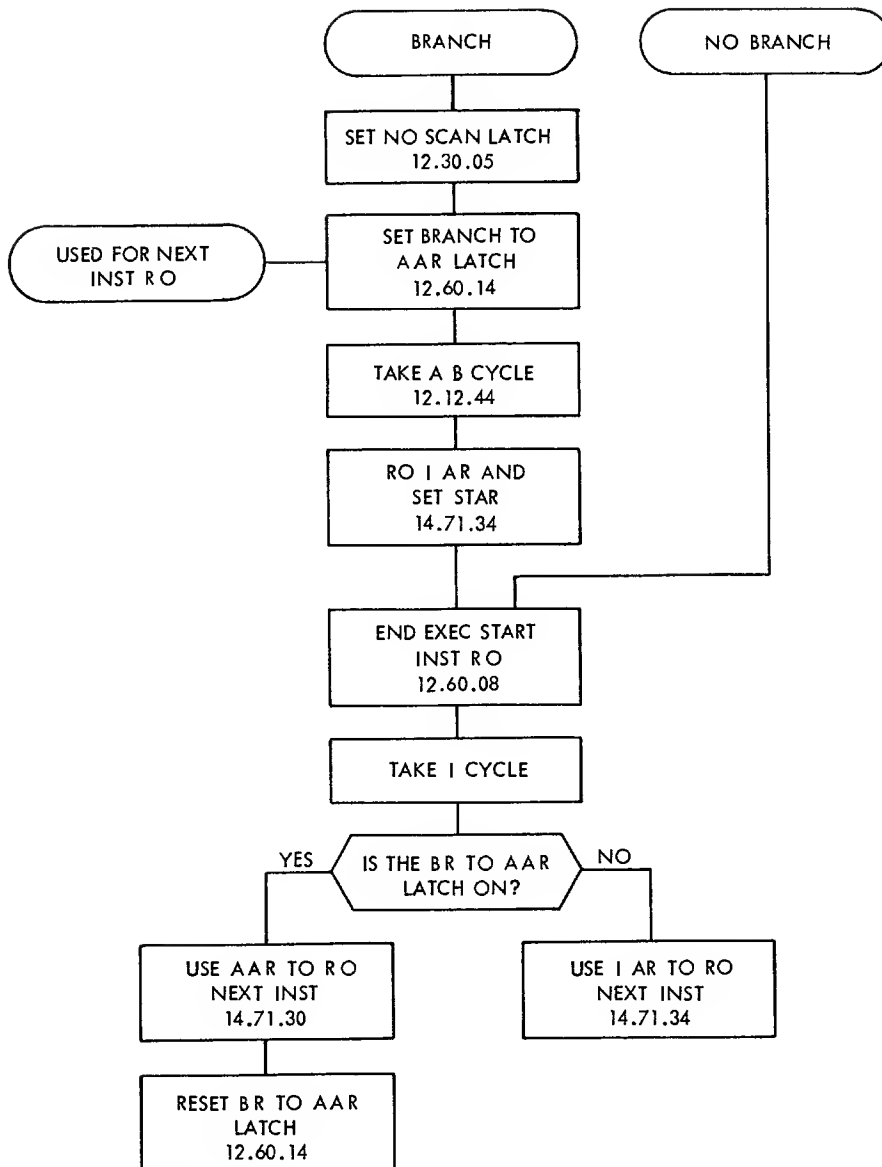


Figure 2.10-1 Branch

d - Modifier	Indicator	Logic
Blank	Unconditional	12.60.01
9	Carriage Channel 9	12.60.01
@	Carriage Overflow (Channel 12)	12.60.01
/	Compare Unequal	12.60.01
S	Compare Equal (B = A)	12.60.01
T	Compare Low (B < A)	12.60.01
U	Compare High (B > A)	12.60.01
V	Zero Result	12.60.01
W	Divide Overflow	12.60.01
Z	Arithmetic Overflow	12.60.02
1	Overlap in Process on Channel 1	12.60.15
2	Overlap in Process on Channel 2	12.60.15
R	Printer Carriage Busy	12.60.15
Q	Branch Inquiry	12.60.02

Figure 2.10-2 Branch Conditions for J Op Code

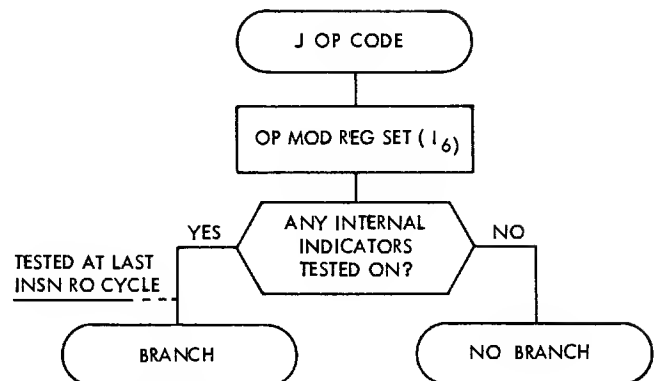


Figure 2.10-3 J Op Code

Op-code grouping lines are:

Instruction Read-out  
Not % type op code  
Addr dbl op code  
I-addr plus mod op codes  
Addr type op codes  
No C- or D-cy op codes

Operational  
Branch type op codes

Control  
Op mod to A-ch on B-cycle ops  
Regen mem on B-cy op codes  
Stop at F on B-cycle op codes  
Read out BAR on scan B-cy ops

Operation (Figure 2.10-3)

At I-ring 6 time during instruction read-out, the Op-modifier register is set with the Op-modifier character and decoded.

The Op-modifier decode is compared with the indicators, and at I-ring 7 time (Last Insn RO Cy) either the branch or no-branch condition is brought up. No-branch causes Last Execute Cycle that initiates instruction read-out of the NSI. Branch causes the extra B-cycle as covered in section 2.10.01 Logic.

#### 2.10.03 Branch if I/O Channel-Status Indicator on (R or X)

##### Op-Code Function

The R or X Op code allows the program to test the I/O channel-status indicator for channel 1 and channel 2 respectively. The d-character modifier specifies the indicator to be tested.

The group-mark modifier (‡) consists of all the BCD bits and tests for any status indicators that are ON. This modifier must be given after each I/O command on each channel. Figure 2.10-4 shows the d-modifiers and the indicators that they test.

Op-code grouping lines are:

Instruction Read-out  
Not % type op codes  
Addr dbl op codes  
1 addr plus mod op codes  
Addr type op code

Operational  
Branch type op codes

d - Modifier	Indicator	Logic	
		Ch. 1	Ch. 2
1	I/O Not Ready	12.60.02	12.60.15
2	I/O Channel Busy	12.60.02	12.60.15
4	I/O Data Check	12.60.02	12.60.15
8	I/O Condition	12.60.02	12.60.15
A	I/O No Transfer	12.60.02	12.60.15
B	I/O Wrong Length Record	12.60.02	12.60.15
$\neq$	Any I/O Channel Status is On	12.60.02	12.60.15

Figure 2.10-4 Branch Conditions for R or X Op Code

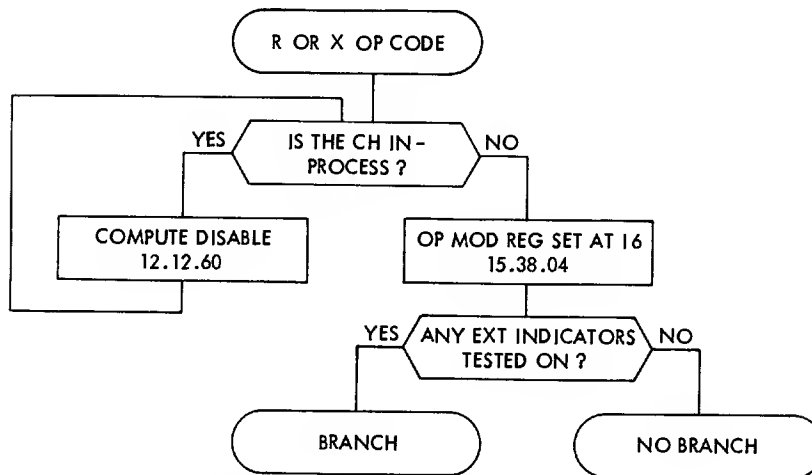


Figure 2.10-5 R or X Op Code

#### Control

Op mod to A-ch on B-cycle ops  
Regen mem on B-cycle op codes  
Stop at I on B-cycle op codes  
Read out BAR on scan B-cycle ops

Operation (Figure 2.10-5)

The R and X Op code are similar in operation to the J-Op code except that at I-ring 6 time, if the channel to be tested is still in-process, the disable compute cycle is brought up to stop the CPU clock and to prevent testing the indicators until the operation is completed.

#### 2.10.04 Branch if Character Equal (B)

##### Op-Code Function

The B-Op code allows the program to compare any character in storage with the d-modifier to determine whether they are the same. The instruction contains the address of this character in the B-field.

Op-code grouping lines are:

#### Instruction Read-out

Not % type op codes  
Not addr dbl op codes  
2 Addr plus mod op codes  
2 Addr op codes  
Addr type op codes  
No C- or D-cycles op codes  
No D-cycle at I-ring 6 ops

#### Operational

Compare type op codes  
Branch type op codes

#### Control

1st scan first op codes  
B-cycle first op codes  
Op mod to A-ch on B-cycle ops  
Regen mem on B-cycle op codes  
Read out BAR on scan B-cycle ops  
Stop at H on B-cy ops

Operation (Figure 2.10-6)

The operation starts by taking a B-cycle to read out the character at the B-field address. The Op modifier register is gated to the A-channel, and a comparison between the A- and B-channels is made in the compare unit. If the equal latch is set, the program branches to the I-address in the AAR. In all other cases the program continues to the NSI.

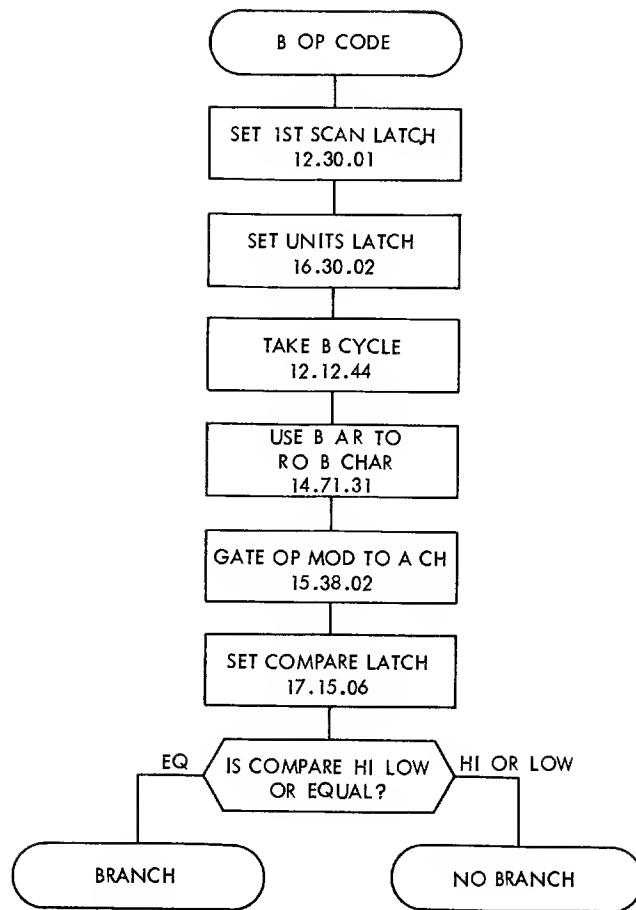


Figure 2.10-6 B Op Code

Bits	d - Modifier	Indicator	Logic
1	1	Word Mark	12.60.03
2	2 (No Zone)	Zone Equal	12.60.03
	B (12 Zone)	Zone Equal	12.60.03
	K (11 Zone)	Zone Equal	12.60.03
	S (0 Zone)	Zone Equal	12.60.03
	3 (WM or No Zone)	WM or Zone Equal	12.60.03
1 & 2	C (WM or 12 Zone)	WM or Zone Equal	12.60.03
	L (WM or 11 Zone)	WM or Zone Equal	12.60.03
	T (WM or 0 Zone)	WM or Zone Equal	12.60.03

Figure 2.10-7 Branch Conditions for U Op Code

### 2.10.05 Branch if Bit Equal (W)

#### Op-Code Function

This instruction causes the character at the B-address to be compared, bit by bit, with the d-character. If any bit in the character at the B-address matches any bit in the configuration of the d-character, the program branches to the I-address.

Op-code grouping lines are:

Instruction Read-out  
Not percent type op codes  
Not addr dbl op codes  
2 Addr plus mod op codes  
2 Address op codes  
Address type op codes  
No C- or D-cy op codes  
No D-cy at I-ring 6 ops

Operational  
Branch type op codes

Control  
1st scan first op codes  
B-cy first op codes  
Op mod to A-ch on B-cy ops  
Regen mem on B-cy op codes  
RO BAR on scan B-cy ops  
Stop at H on B-cy ops

Operation (See Figure 2.10-6)

The operation is the same as the B-Op code except that individual bits are compared to determine whether a branch should be taken (12.60.03).

### 2.10.06 Branch on Word Mark or Zone Equal (V)

#### Op-Code Function

The character at the B-address is examined for the specific zone or word-mark combination specified by the d-modifier (Figure 2.10-7). If the condition is satisfied, the program branches to the I-address; otherwise, the program continues to the NSI.

Op-code grouping lines are:

Instruction Read-out  
Not % type op codes  
Not addr dbl op codes  
2 Addr plus mod op codes  
2 Addr op codes



Addr type op codes  
No C- or D-cycles op codes  
No D-cycle at I-ring 6 ops

Operational

Branch type op codes

Control

1st scan first op codes  
B-cycle first op codes  
Op mod to A-ch on B-cycle ops  
Regen mem on B-cycle op codes  
Stop at H on B-cycle op codes  
Read out BAR on scan B-cycle ops

Operation (Figure 2.10-8)

This operation duplicates the B-Op code until the character that is designated by the BAR is read out to the B-channel. At this point, the zones on the B-channel are compared with the zones of d-modifier (gated to the A-channel on this B-cycle) to determine if they are the same.

A branch is initiated, if the d-modifier contains a 2-bit and the zones compare equal, or if the modifier has a 1-bit and there is a B-channel WM.

## 2.11.00 MISCELLANEOUS OP CODES

### 2.11.01 Table Look-Up (T)

#### Op-Code Function

The T-Op code causes the machine to search through a table to locate a table argument that is equal to, lower than, or higher than the search argument as determined by the d-modifier (Figure 2.11-1).

The A-field address locates the search argument. The B-field address locates the low-order character of the table.

An example of table search is locating the square of a number (search argument). A table of squares is stored in the core-storage unit. The table includes the number (table argument), followed by the square of the number (function). See Figure 2.11-2 that contains square arguments and functions. The end of the table is designated by a B-field shorter than the A-field. In the case of a single-character A-field, the condition specified by the modifier must be met to stop the operation.

Common Op-code grouping lines are:

#### Instruction Read-out

Not percent type op codes  
Not addr dbl op codes  
2 Addr plus mod op codes  
Two address op codes

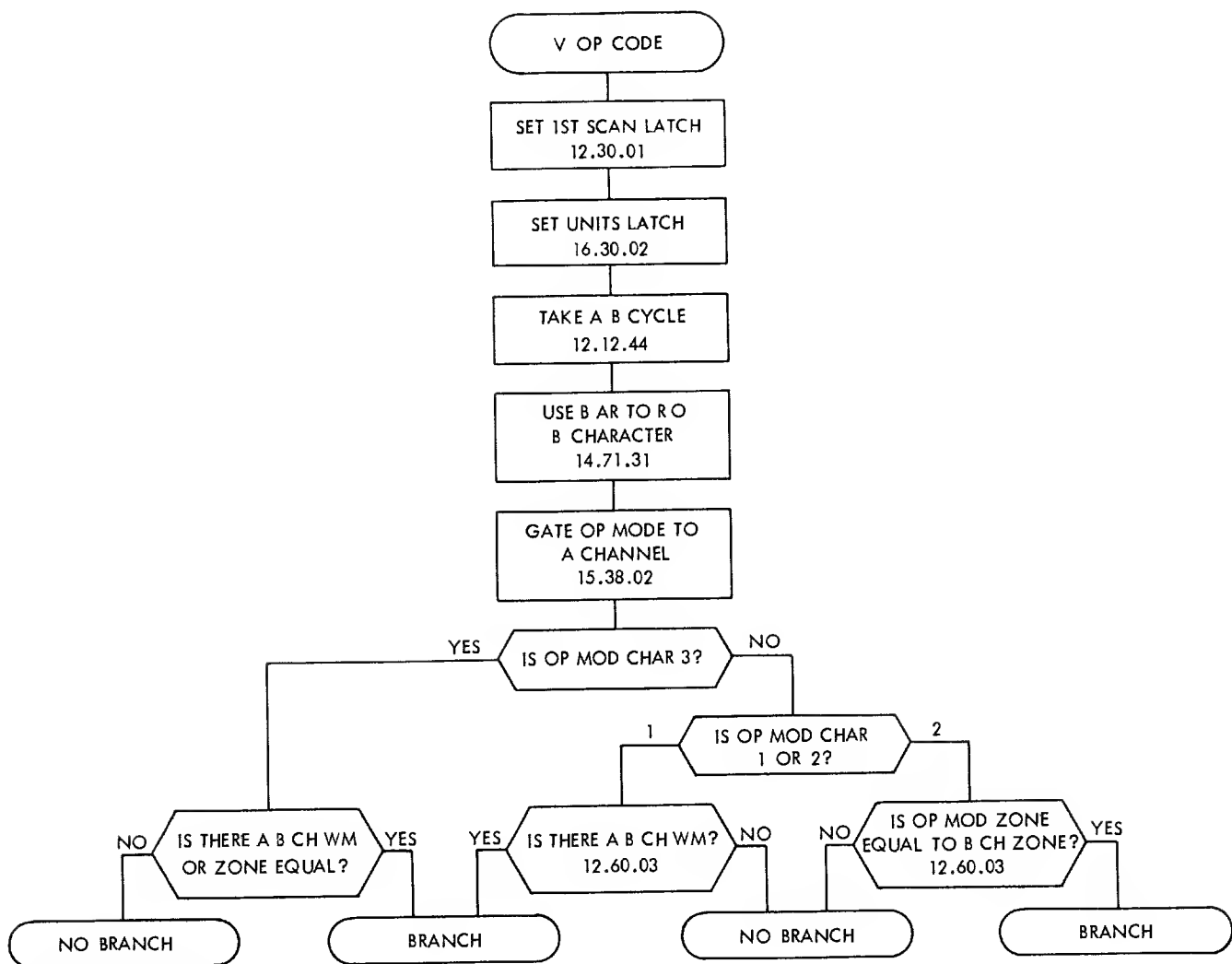


Figure 2.10-8 U Op Code

Addr type op codes  
C-cycle op codes  
No D-cycle at I-ring 6 ops

Operational

Compare type op codes  
No-branch op codes

Control

1st scan first op codes  
A-cy first op codes  
Std A-cycle op codes  
A-reg to A-ch on B-cycle ops  
Regen mem on B-cy op codes  
Read out BAR on scan B-cy ops

Operation (Figure 2.11-3)

During the Last Insn RO cycle the controls are set for an A-cycle with 1st scan control. The units latch is set ON and causes the CAR to read out the units position of the A-field (search argument). A B-cycle is initiated. At that time BAR reads out the units position of the B-field (table argument).

The A-channel and B-channel characters are compared, and the result sets the high, low, or equal compare latch. The absence of an A-Ch or B-Ch WM initiates another A-cycle and the AAR is used to read out the tens position of the search argument. A B-cycle follows during which the BAR is used to read out the tens position of the table argument.

Character-by-character comparison continues by means of alternate A- and B-cycles, until the A-Ch WM is reached. When the A-Ch WM is encountered, the compare latch setting is analyzed. If the latch setting satisfies the condition that is specified by the Op modifier, the table look-up operation ends. If the condition is not met, B-cycles are taken successively until a B-Ch WM is encountered. This indicates the end of the function field. The CAR retains the location of the units position of the A-field and permits the search argument to be compared with the next B-field table argument. Comparison of the search argument with the table arguments continues until the condition of the Op modifier is met, or until a B-Ch WM is encountered before an A-field WM (that is, until a B-field shorter than an A-field is encountered). In the latter event, the high latch is set ON, and the table look-up operation terminates.

## 2.11.02 Store Address Register (G)

### Op-Code Function

The contents of the register that is specified by the d-character are stored in the

DESCRIPTION	d CHARACTER	BIT CONFIGURATION	STOP IF TABLE ARGUMENT IS:
LOOKUP EQUAL	2	2	EQUAL TO SEARCH ARGUMENT
LOOKUP LOW	1	1	LOWER THAN SEARCH ARGUMENT
LOOKUP HIGH	4	4	HIGHER THAN SEARCH ARGUMENT
LOOKUP LOW OR EQUAL	3	2 1	EQUAL TO OR LOWER THAN SEARCH ARGUMENT
LOOKUP EQUAL OR HIGH	6	4 2	EQUAL TO OR HIGHER THAN SEARCH ARGUMENT
LOOKUP LOW OR HIGH	5	4 1	LOWER THAN OR HIGHER THAN SEARCH ARGUMENT

Figure 2.11-1 d-Character for Table Lookup

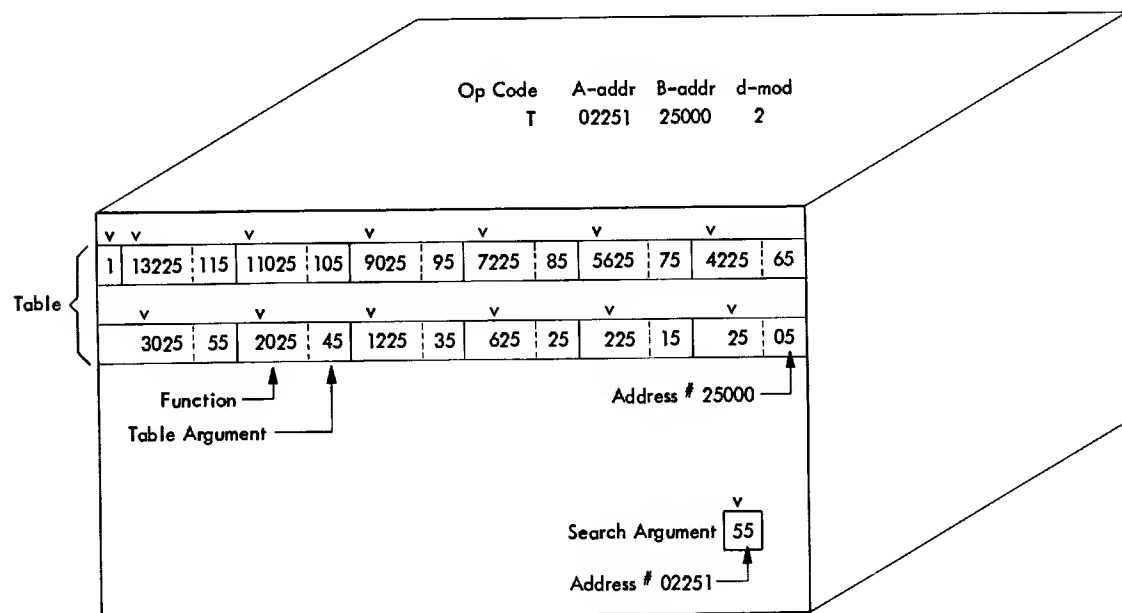


Figure 2.11-2 Table Lookup Operation

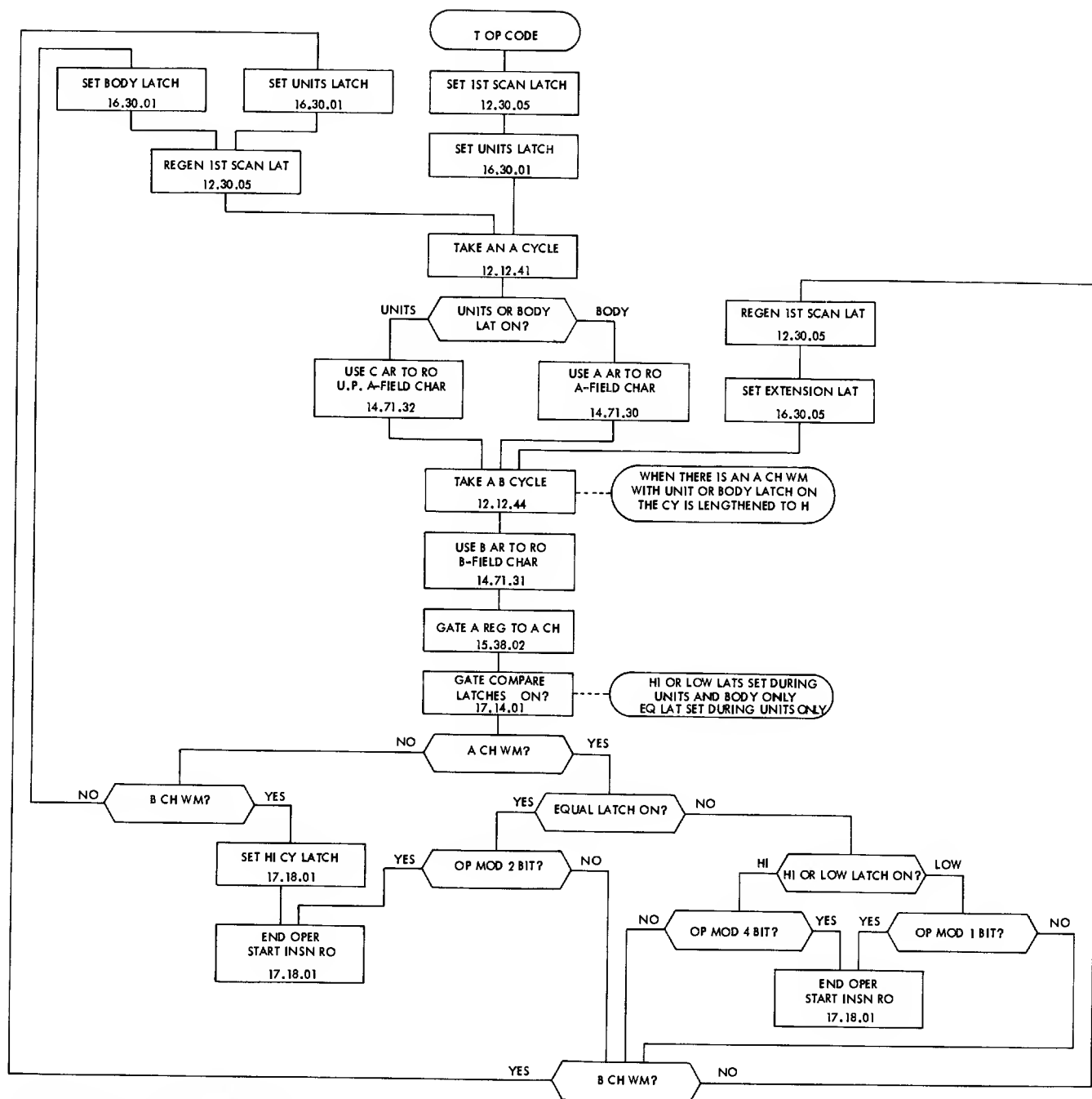


Figure 2.11-3 Table Lookup Op Code

C-field. The C-address specifies the low-order position of the field in which the register contents are to be stored.

#### Op-Modifier Function

The d-character specifies the register to be stored, as follows:

<u>d-Character</u>	<u>Operation</u>
A	Store A-address register
B	Store B-address register
E	Store E-address register
F	Store F-address register

Common Op-code grouping lines are:

Instruction Read-out  
 Not addr dbl op codes  
 1 Addr plus mod op codes  
 Addr type op codes  
 No index on 1st addr ops

Operational  
 No-branch op codes

Control  
 1st scan first op codes

#### Operation (Figure 2.11-4)

A series of 5 C-cycles takes place. During each C-cycle, the CAR is modified by minus one. The A-ring advances through A6 serially to gate each position of the selected address register out to the address exit channel. The contents of the address register that is being stored is then switched to the A-data register, to the A-channel, and through assembly to the storage location addressed by the CAR. The operation ends after the ten-thousands position of the selected address register is stored at A-ring 6 time.

#### 2.11.03 Set Work Mark (,)

##### Op-Code Function

The comma (,) Op code causes a word mark to be set at each address that is specified in the instruction. The data at each address is undisturbed. If this instruction is given with one address (A-address), a word mark is set at the A-address only. If the address instruction is indexed, a word mark is set at the locations that are specified by the indexed A-address. However, if this instruction is given with no A-address (no-address instruction), word marks are set at the locations that are specified by the addresses in both the A- and B-address registers (the contents from the previous operation).

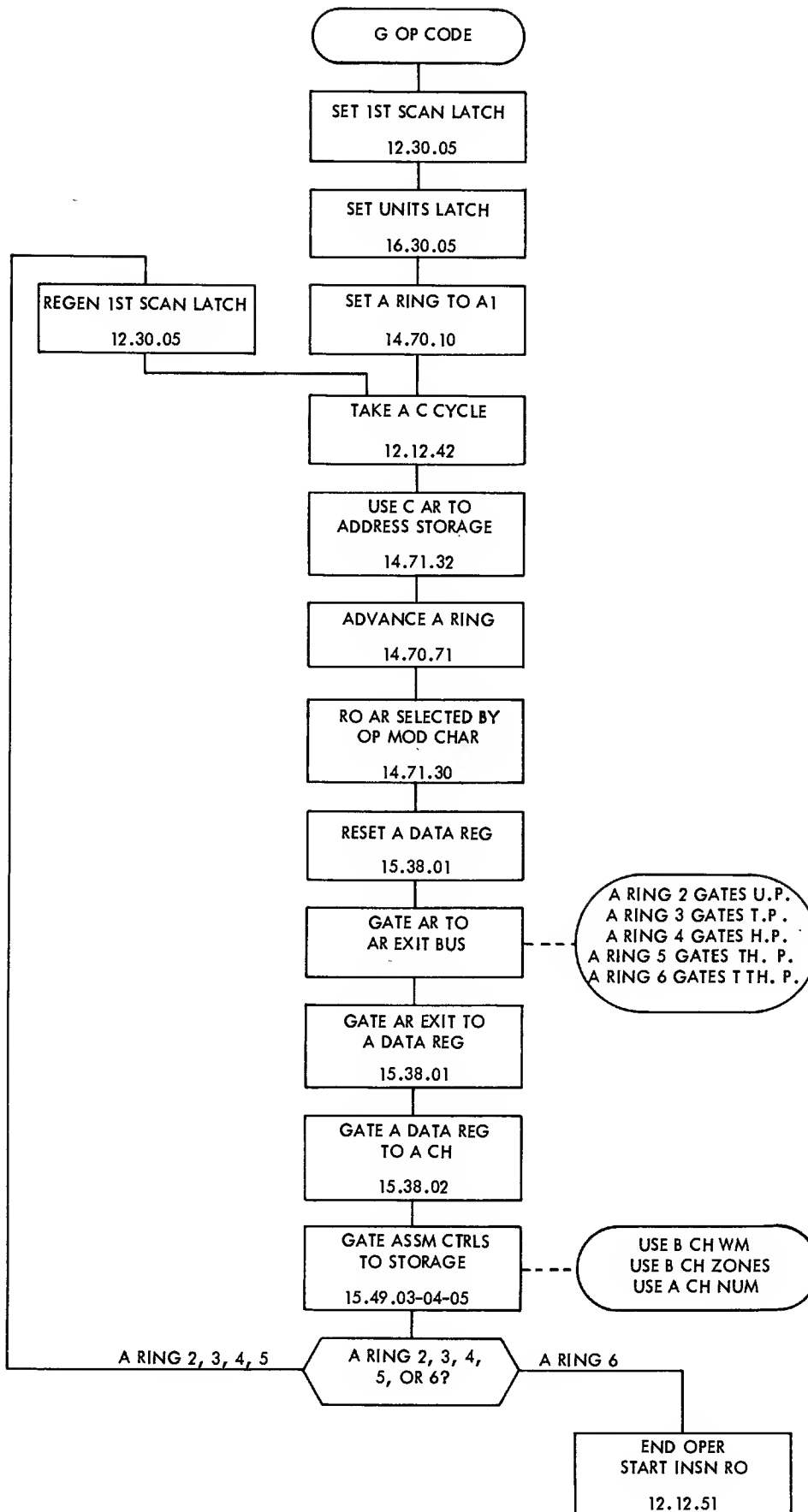


Figure 2.11-4 Store Address Register

Op-code grouping lines are:

Instruction Read-out  
Not percent type op codes  
Addr double op codes  
2 Addr no mod op codes  
Two address op codes  
Addr type op codes  
No C- or D-cy op codes  
No D-cy at I-ring 6 ops

Operational  
No-branch op codes  
Word mark op codes

Control  
1st scan first op codes  
A-cy first op codes  
A-reg to A-ch on B-cy ops  
Read out BAR on scan B-cy ops  
Read out AAR on A-cy ops

Operation (Figure 2.11-6)

One A- and one B-cycle are required to execute this operation. During A-cycle, the A-address character reads from storage to the B-channel. It is then gated through the assembly and back into storage accompanied by a WM bit. Also at this point, a check bit is added or removed to maintain parity.

During B-cycle, the character at the B-address is read from storage and treated similarly. If only one address is specified, the A-data address is stored in both the A- and B-address registers during instruction RO. Therefore, when execute phase begins for a single-address instruction, the A-data address is in both the AAR and BAR. Thus, a WM is set in the same location twice, once during the A-cycle, and once during the B-cycle.

If this instruction is given with no address, WM's are set in A- and B-address locations that are specified in the previous instruction. The operation is the same to a two-address instruction.

#### 2.11.04 Clear Word Mark (⌘)

##### Op-Code Function

The ⌘ Op code causes word marks to clear from the locations specified by the A- and B-addresses of the instruction. The data of each address is undisturbed. If this instruction is given with one address, a word mark is cleared at the A-address only. If the instruction is indexed, a word mark is cleared at the location that is specified by the indexed address. However, if this instruction is given with no addresses, word marks are cleared at the locations that are specified by the addresses in both the A- and B-address registers from the previous operation.



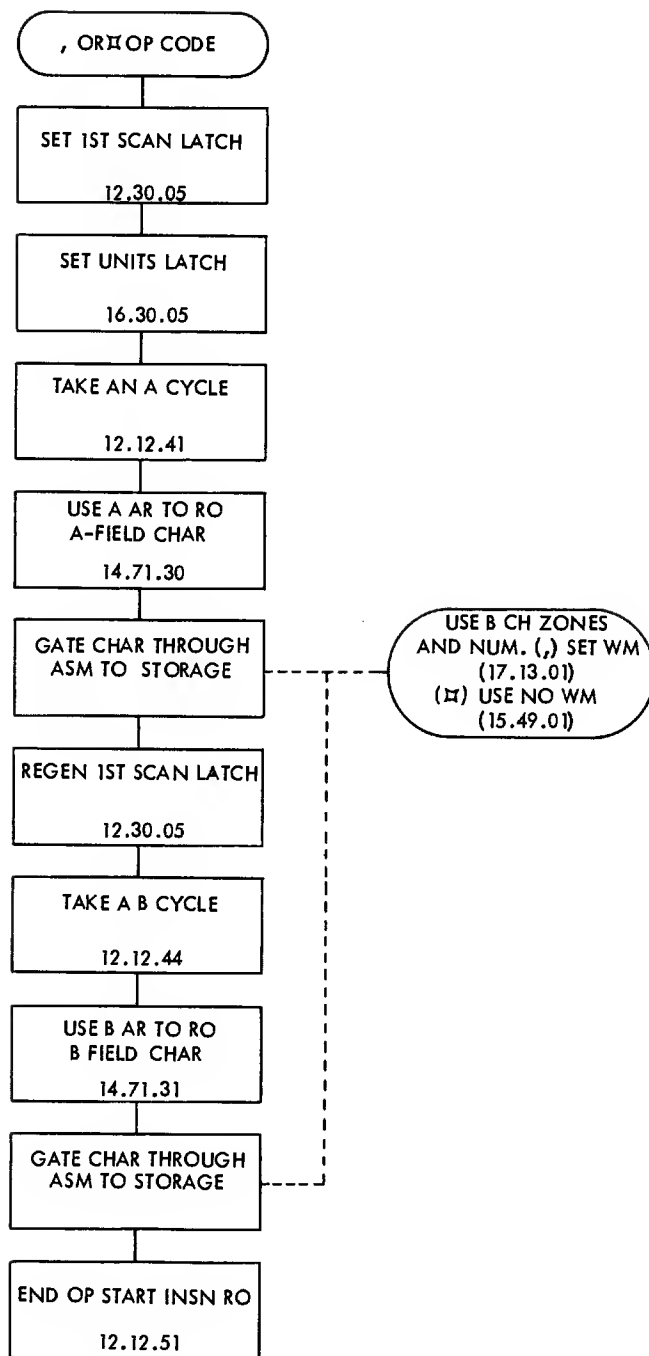


Figure 2.11-5 Set or Clear Word Marks

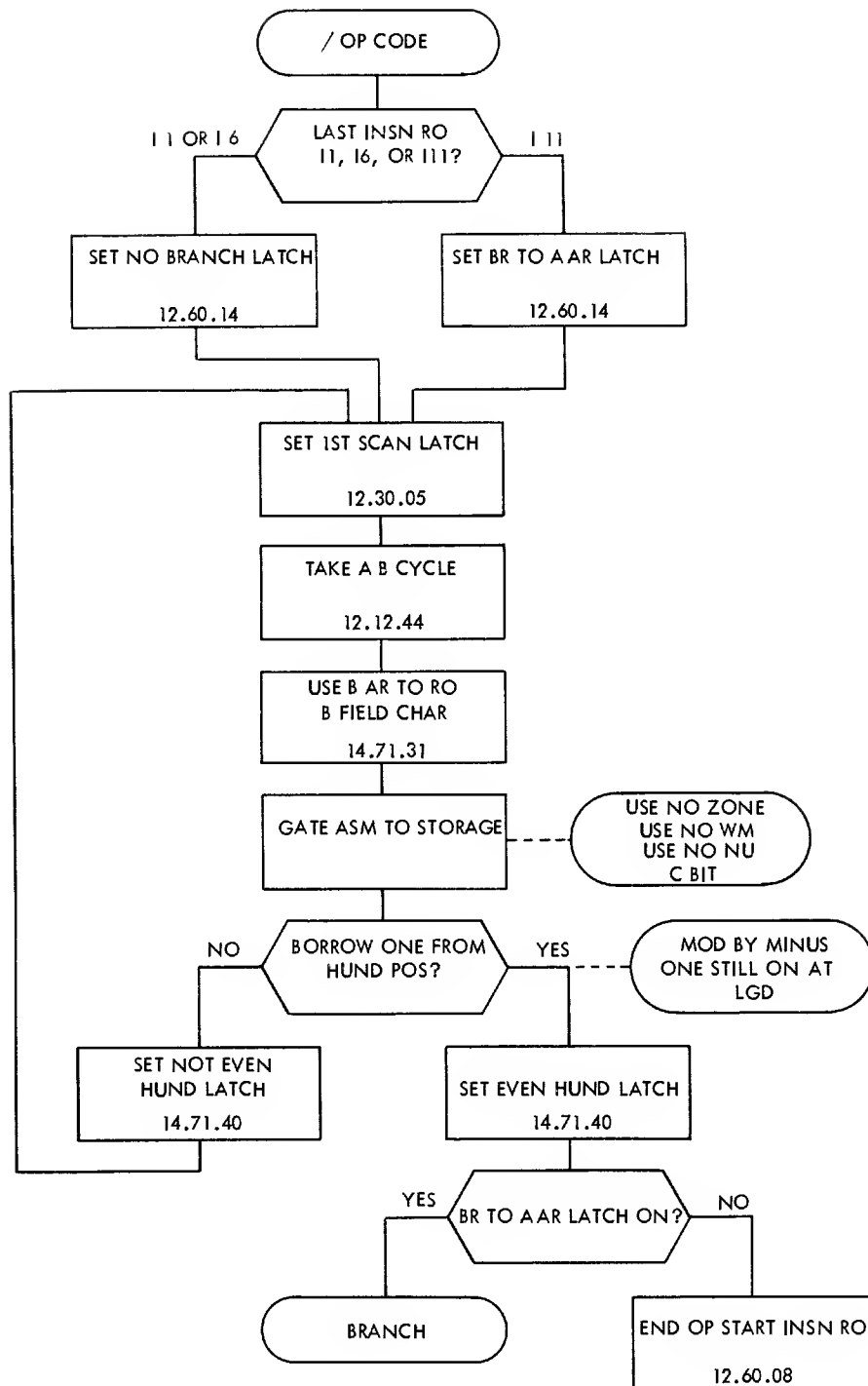


Figure 2.11-6 Clear Storage

Common Op-code grouping lines are:

Instruction Read-out  
Not percent type op codes  
Addr dbl op codes  
2 Addr plus no mod op codes  
2 Address op codes  
Addr type op codes  
No C- or D-cy op codes  
No D-cy at 1-ring ops

Operational  
No-branch op codes  
Word mark op codes

Control  
1st scan first op codes  
A-cy first op codes  
A-reg to A-ch on B-cy ops  
RO BAR on scan B-cy ops  
RO AAR on A-cy ops

Operation (Figure 2.11-5)

This operation is executed in the same manner as set word mark, section 2.11.03. One A- and one B-cycle are taken. The character to be treated is handled in the same manner. The same Op-code grouping lines are activated. The significant difference is that the WM bit is removed instead of set, as the character is gated through assembly to storage.

#### 2.11.05 Clear Storage (/)

##### Op-Code Function

The / Op code causes as many as 100 positions of core storage to be cleared of data and word marks. Clearing starts at the A-address and continues leftward to the nearest hundreds position. The cleared area is set to blanks.

Op-code grouping lines are:

Instruction Read-out  
Not percent type op codes  
Addr double op codes  
2 Addr no mod op codes  
Two address op codes  
Addr type op codes  
No C- or D-cy op codes  
No D-cy at 1-ring 6 ops

Operational  
Branch type op codes

Control  
1st scan first op codes  
B-cy first op codes  
A-reg to A-ch on B-cycle ops  
Stop at F on B-cycle op codes  
Read out BAR on scan B-cy ops

Operation (Figure 2.11-6)

The operation is executed by a series of B-cycles during which one address location in storage is cleared. Clearing starts with the location that is specified by the A-address, and continues leftward (address modifier set to -1) to the nearest even-hundreds position. The minus-one condition ON at logic gate D-time defines the even-hundreds address. This indicates a borrow-one-from-hundreds position. This sets the even-hundreds latch and ends the execute phase.

When this instruction is given without an address, the contents of the B-address register from the previous operation are used as the A-address. In this case the A-address register is not loaded at Ins RO time, and is undisturbed at the end of the clear storage operation.

Clear Storage and Branch (/)

Op-Code Function

This Op code performs the same operation as / (A) except that the clearing starts at the B-address. The location of the next instruction is specified by the I-address. This is an unconditional branch.

Operation (Figure 2.11-6)

As described for the Clear Storage operation, Clear Storage and Branch is executed by a series of B-cycles. Because the last instruction RO cycle occurs at I-ring 11 time, however, the branch to AAR latch is set ON. When the even-hundreds address is set ON and gated with the branch to AAR latch ON, No Scan Control is set and another B-cycle is taken. This causes the next instruction to be taken from the address that is specified by the AAR as described in the Branch Operation (section 2.10.01).

2.11.06 Half ( . )

Op-Code Function

The ( . ) Op code causes the machine to stop, and initiates a stop-print-out operation. Pressing the start key causes the program to start at the next instruction in sequence.

Op-code grouping lines are:

Instruction Read-out  
Not percent type op codes  
Addr type op codes  
No C or D-cycle op codes  
No D-cy at I-ring 6 ops  
WM, E, Z, W, V, C, Clear or . ops

Operational  
Branch type op codes

Control  
A-reg to A-ch on B-cycle ops  
Regen mem on B-cy op codes  
Stop at F on B-cycle op codes  
Read out BAR on scan B-cy ops

#### Operation

The last instruction read-out cycle (determined by the presence of a B-Ch WM bit at I-ring 1-time) causes a simultaneous-last-execute cycle. This sets the stop latch, stops the logic clock, and initiates a stop-print-out operation. See the section 7.1.09 entitled Stop Print-Out.

#### Halt and Branch . (I)

#### Operation

At the end of the instruction-read-out cycle, AAR is set with the address in storage to which you desire the machine to branch for its next instruction.

A B-cycle is then initiated during which the IAR reads out to the STAR and through the modifier (set to Modify by Zero Condition) to the BAR. The last execute cycle stops the logic clock and initiates the stop print-out.

The Br to AAR latch is set to cause the AAR to read out to STAR for the I-Op cycle, during instruction read-out when the start key is pressed.

#### 2.11.07 No Operation (N)

#### Op-Code Function

The N-Op code performs no operation. It can be substituted for the operation code of any instruction to make that instruction ineffective.

#### Operation (Figure 6.2-1)

During I-Op time, the N-Op from Op Decode prevents I-Ring Advance. This results in a succession of I-cycles. The address modifier is set to Plus 1, and I-cycle control causes the storage to continue to read out characters until another Op code is detected. A B-channel WM causes the next Op code to be set into the Op-reg and the instruction read-out continues.

### 3.0.00 INTEGRATED SYNCHRONIZER

The integrated synchronizer is an input-output controlling device that is capable of transferring data between the Central Processing Unit (CPU) and the associated input-output equipment. The equipment can consist of the IBM 1402-2 Card Read-Punch, and up to six optional serial-data devices such as the IBM 1011 Paper Tape Reader. All units operate on the efficient memory-sharing basis.

Input-output transfers are brought about by using magnetic core-storage units. The capacity of each unit is 80 characters. A common addressing scheme is shared by all core-storage units. Information is stored in the Binary Coded Decimal (BCD) form. Each storage position can store any of the 64 valid characters represented by combinations of the six possible data bits and the check bit (C, B, A, 8, 4, 2, 1).

Use of the integrated synchronizer results in saving processing time in the CPU. For example, on a card-read operation, the CPU continues processing while the card is read into the reader storage unit. When the read-in is complete, the reader becomes not-busy. This condition signals the CPU that the card information is available for transfer. When processing is stopped to transfer the card data, the transfer takes place in a maximum of several milliseconds. This is far less time than it would require to transfer directly from the card.

The same principle holds true for all integrated synchronizer operations, whether input or output, parallel or serial.

In addition to saving CPU time, the integrated synchronizer allows any or all units associated with it to run simultaneously.

#### 3.1.00 CARD PUNCH

The 1410 system executes a card punch command in several steps. First, the information to be punched transfers from the CPU core-storage unit to the punch storage unit. The CPU reads out in reverse scan, and places the characters in the E-data registers. Punch storage accepts the characters and stores them in sequence in BCD form. Two rings (units and tens) are used to address the 80 positions of punch storage. The 80 positions are labelled 00-79 and cause punching into card columns 01-80. Each character is checked for correct odd parity as it enters punch storage.

Once a correct transfer of 80 characters is made, the punch clutch in the 1402 is energized. This causes a card to feed under the punches face down, 12-edge first. Just before the 12-row of the card reaches the punching station, all 80 positions of punch storage must be scanned out in order, and decoded to see whether any 12 punches are to be punched (Figure 3.1-1). For each card column that is to have a hole punched at this time, the punch magnet that corresponds to that column is energized. When the actual punching takes place after the scan, those columns are punched for which magnets are energized. After the punching takes place the magnets are de-energized as the card continues to feed through the punch feed.

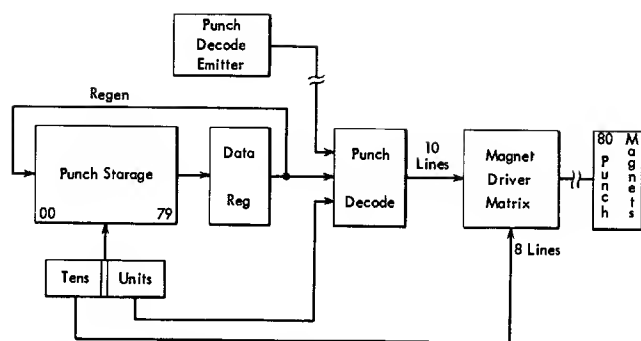


Figure 3.1-1 Punch-Scan Data Flow

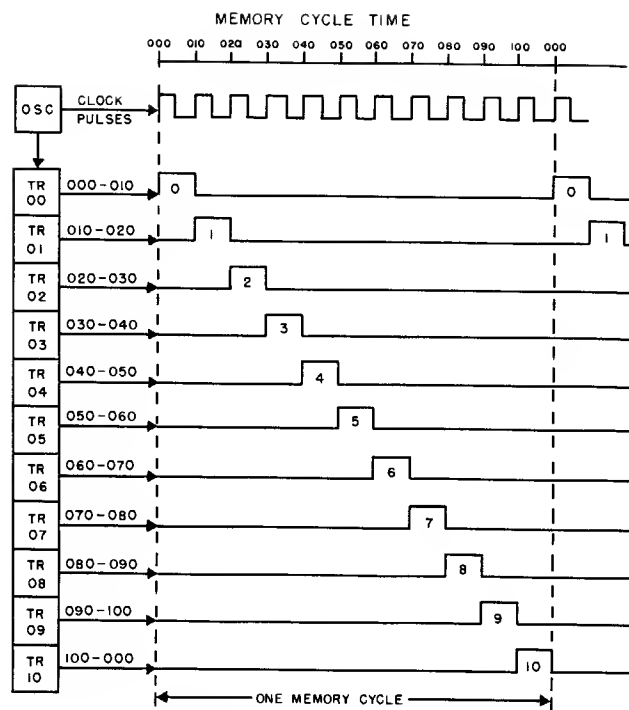


Figure 3.1-2 Clock Logic

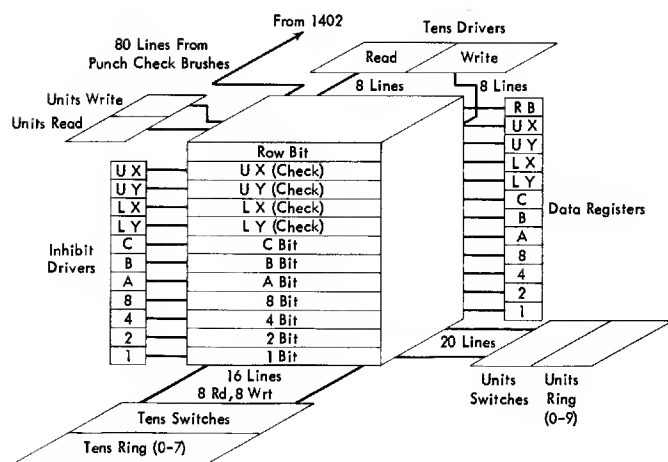


Figure 3.1-3 Punch Storage

During the 12-row scan, every position is read out in search of positions to be punched with a 12. To retain the information in storage so that the rest of the card rows can be punched, the information in each position is regenerated.

Just before the 11-row in the card reaches the punches, another scan of all 80-punch storage positions is taken in search of those positions that are to have an 11-hole punched in them. The proper magnets are energized and the 11-row is punched. Once again, each position is regenerated after it reads out. The same sequence of events takes place for each of the remaining rows in the card.

Included in the punch storage circuitry is a punch-hole-count check. This check is used to determine whether the punch out is correct. The check consists of a comparison between the number of holes that should be punched in each column and the number of holes actually punched. The comparison requires two sources of information. The first source is a bit-generator circuit that determines the number of holes to be punched in each column. This information (number of holes) is developed while the card is being punched, and is sent to the hole-count check circuits where it is stored. During the following card feed cycle, the punched card passes under a set of punch check brushes where the actual number of holes that are punched is determined. This information also goes to the hole-count check circuit for comparison with the information stored there on the previous card feed cycle.

After the check is completed, the card feeds out to one of three stackers that is selected by a character in the card-punch instruction word.

### 3.1.01 Clock

All pulses and gates that the integrated synchronizer requires are produced by a clock circuit. A free-running, one-megacycle oscillator develops one clock pulse every microsecond. This clock pulse drives an eleven-position closed ring. It takes eleven pulses or eleven microseconds to advance this ring completely around. The eleven microseconds that this requires is defined as a memory cycle (Figure 3.1-2). To address and to read a character in or out of a storage position takes one memory cycle.

Timings are expressed as zero through ten time (00.0 - 10.0). Note: On ALD's, decimals are not shown.

### 3.1.02 Core Storage (Figure 3.1-3)

Punch storage consists of an 80-position, 12-plane, core-storage unit. Seven planes store the seven BCD bits for each position. Four planes (upper X, lower X, upper Y, and lower Y) are used by the hole-count check circuits. An additional plane, the row-bit-storage plane, is an intermediate storage device that transfers data between the punch check brushes in the 1402 and the hole-count check circuits.

There are separate read and write windings that pass through each data core (Figure 3.1-4). The cores in the BCD and check planes are all considered data cores. To set (read in) a core, half-write current is passed through each write winding. Current through the inhibit winding opposes one of the half-write windings to prevent setting the core.





Information contained in the cores is lost when the read-out takes place. The coincidence of the half-read currents causes the core to flip to the opposite state from which it is set with the write currents. When the core flips, a pulse is induced onto the sense winding. This pulse is amplified and is used to set a latch, called the data register.

The punch-row bit cores (Figure 3.1-4) are set in a parallel manner. When the brush impulse CB makes, the brush-driver transistor causes current to flow through the half-write winding that passes through all 80 cores in the row-bit plane (Section 14A, 1402 Wiring Diagram). This half-write current is not enough to flip any core by itself. However, any holes in the card are sensed by the punch brushes. This causes current to flow through the six-turn windings of the corresponding row-bit cores (Figure 3.1-5). Current through the six-turn windings plus the half-write current produces a magnetic field great enough to set any core.

In addition to the six-turn and half-write windings, there are two read windings and a sense winding that read out the row-bit cores. The punch-row bit-read windings are impulsed at the same time as the read windings of the punch data planes. Therefore, the row-bit cores read out along with the data cores. This row-bit information is sent to the hole-count check circuits (Section 3.4.04 Hole-Count Check).

A scan of all 80-storage positions takes place at each cycle point (card row), when a card is feeding through the punch feed. As each position is addressed, all cores for that position are read out. The row-bit core and hole-count check cores are read out for hole-count checking. The data cores are read out for decoding.

### 3.1.03 Punch-Storage Scanning

#### Ring-Addressing Scheme

The punch-core-storage unit consists of ten rows of 8 positions each, or a total of 80 positions. When reading in or out of the unit, all 80 positions are scanned starting with position 00 and going through 79.

Two rings, units and tens, are used to address the core-storage unit. The units ring consists of ten triggers (0-9). Outputs of the triggers drive ten units switches. The units switches select one of ten rows of core positions in one direction. The tens ring consists of eight triggers (0-7). Outputs of these triggers drive eight tens-switches. The tens-switches select one of eight rows of core positions in the other direction. Selecting a row in each direction produces one selected position at the coincidence of the two rows. The selected position can then be either read into or read out of, depending on whether the read or write drivers are impulsed.

By controlling the advance of the rings, the desired storage unit is scanned in a sequence. The units ring advances at the beginning of every memory cycle (Figure 3.1-6). The tens ring advances at the beginning of the memory cycle that follows a units-ring carry. Therefore, the core-storage units are scanned in sequence from position 00 to 79 on successive memory cycles. Notice that each memory cycle consists of reset, read, write, and inhibit timings. On each memory cycle, the data registers are reset, and a position is addressed, read out, and can be written back into.

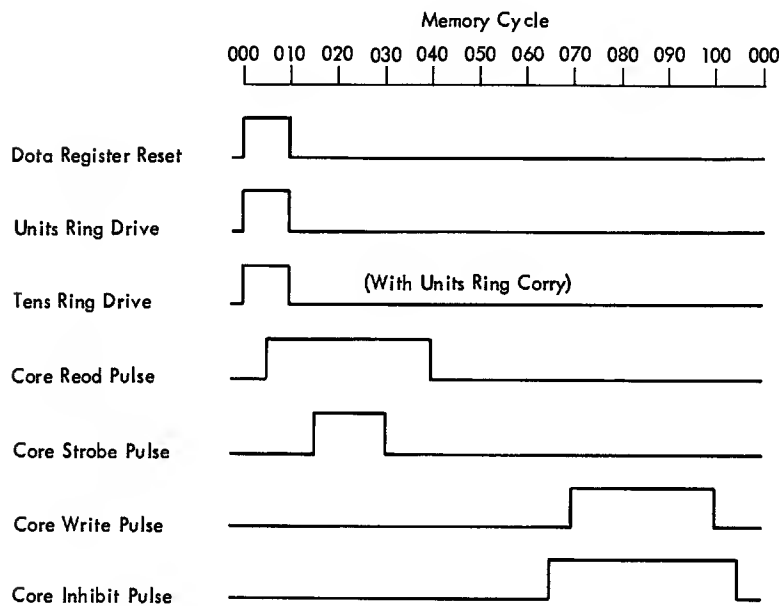
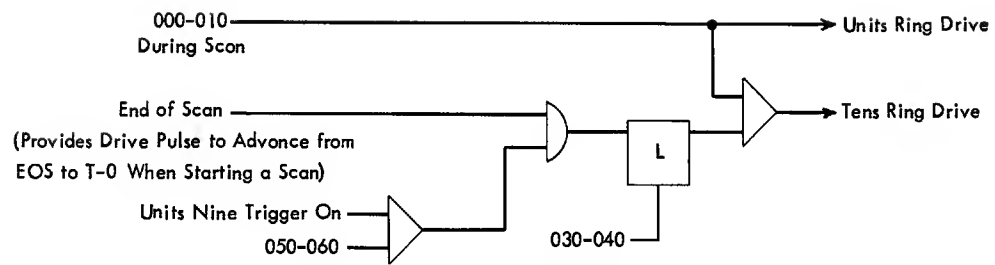


Figure 3.1-6 Ring Drive and Core Timing

For example, to read out of punch-storage position 16, the tens ring is at 1, and the units ring at 6. Impulsing the units and tens read drivers causes current to flow through both read windings of position 16. This flips the cores that are set (contain bits). The flipped cores induce currents on the sense windings to set the proper data-register latches.

Writing in an addressed position involves blocking inhibit current for the cores that are to be set, and allowing the write current to set the cores.

### Punch Priority Controls

Fourteen complete scans are taken to punch a card. A transfer scan fills the storage unit with CPU data. Twelve scans, one for each row in the card, are taken to punch the twelve rows in the card. Another scan completes the hole-count check. A priority control circuit initiates the necessary scans.

The common addressing scheme of the integrated synchronizer allows all core-storage units to be addressed with the same system of rings, switches, and drivers. This means that only one storage unit (punch, reader, etc.) can be scanned at one time. Because more than one I/O unit can be in motion at one time, it is possible for the priority control circuit to receive scan requests from several units simultaneously. When several units request a scan at the same time, the priority control circuit grants the request of the highest priority unit first. A scan request is not granted unless there are no other scans in progress, and unless there are no scan requests with higher priority.

When a punch scan or a punch-transfer scan is requested, the priority conditions must be met before the scan is initiated (Figure 3.1-7). A punch scan is requested whenever a card row is in position to be punched, or when you want to transfer a record from the CPU to punch storage. The thirteen scans that are required to punch and to check a card are requested by a punch scan CB in the 1402. When transferring from CPU to punch storage, the CPU requests the punch scan (Figure 3.1-8).

#### 3.1.04 Punch Decode and Punch Matrix

Information that is stored in the punch-storage unit in preparation for card punching is in BCD form. To punch the information, a punch-decode circuit translates the BCD characters into the IBM card code. A punch emitter signals the decode circuitry at punch-index time so that the punches are set up at the correct time.

For example, when the emitter signals that the punch-feed index is at 12 time, positions containing an A-bit and a B-bit energize corresponding punch magnets.

Punch decoding takes place during each memory cycle as a character is read out of a punch-storage position. The resultant decoded information is sent to the punch matrix where the proper magnet is selected and energized.

The matrix is arranged so that as each core-storage position (00-79) is addressed, the corresponding punch magnet (1-80) is selected. Decode circuitry determines whether the punch magnet is energized.

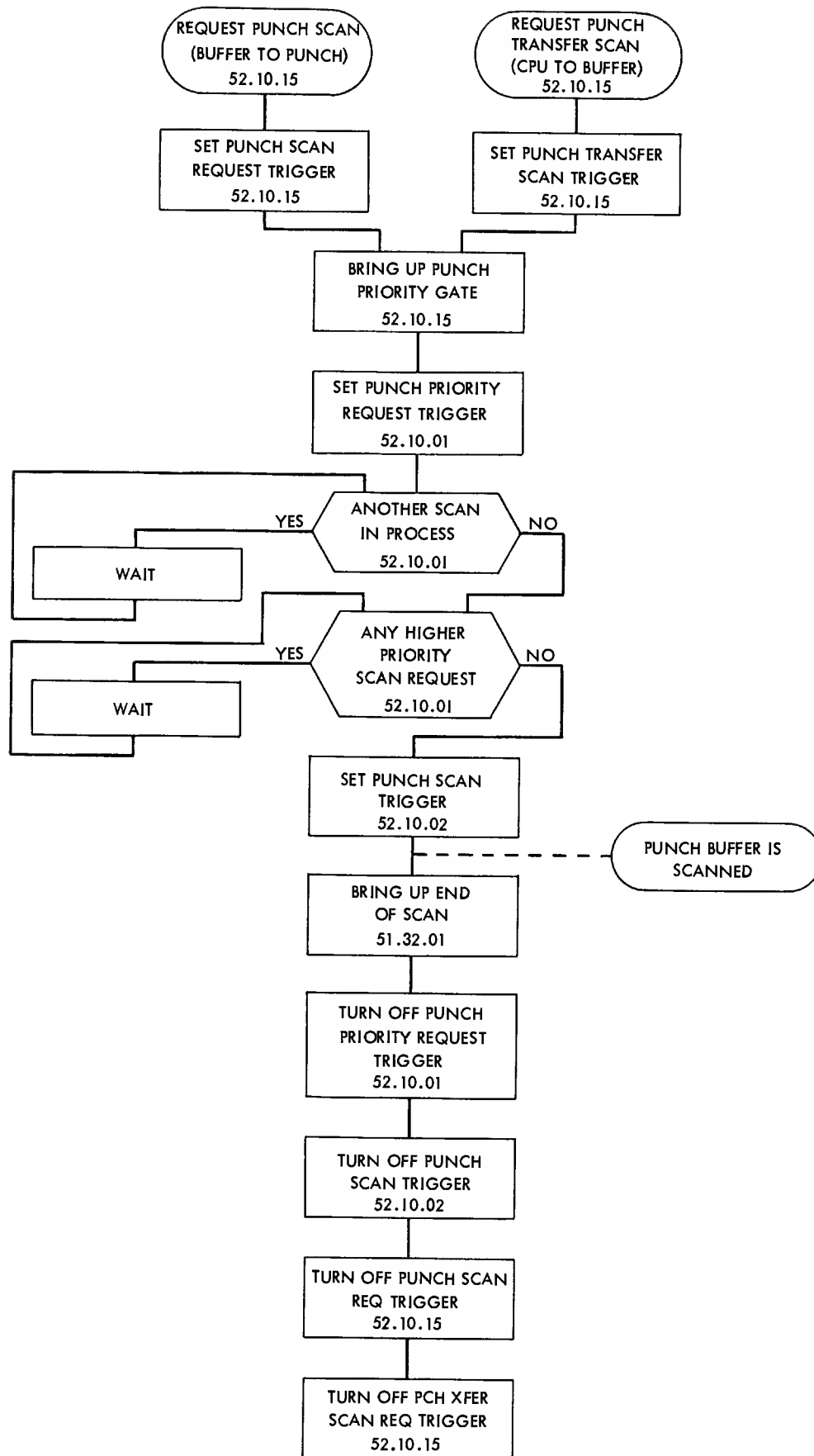
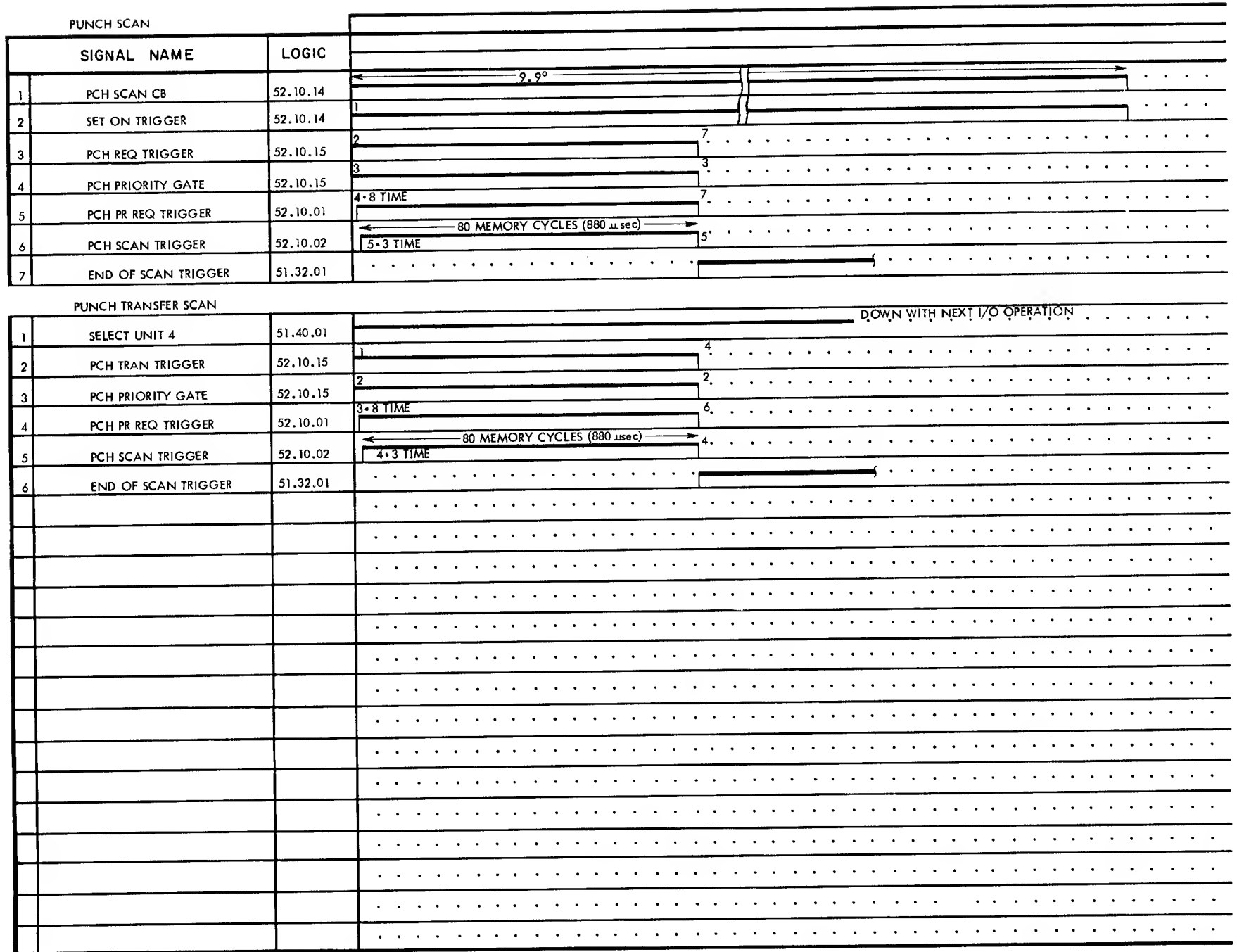


Figure 3.1-7 Card Punch Priority Control

Figure 3.1-8 Punch-Scan Request Timing



### 3.1.05 Punch-Move or Load Op Code

#### Op Code Function (O XXX BBBBB d)

A punch-move or load Op code causes an 80-character record to be transferred from the CPU core-storage unit to punch storage in the integrated synchronizer. If the transfer is made correctly, the punch is set in motion, and the 80-character record in punch storage is punched in a card.

Word marks cannot be handled as such by the card punch because the word mark is not a character. It is a bit that is added over a character in storage. The punch-move Op code transfers data only, and disregards any word marks. The punch-load Op code permits the combination of characters and their associated word marks to be recorded in the punched card. A word mark is punched as a word-separator character in the column to the left of the character that accompanies the word mark in storage. A word-separator character in storage is punched as two successive word-separator characters.

#### Status Indicators

The punch status can prevent the punch command from being executed. For example: the stacker may be full; there may have been an error on the last card punched; or there may be a punch jam. If this sort of situation occurs, it is good to release the CPU for other operations. To do this, the input-output channel uses six status-indicator latches. The indicators are set to define the status of the addressed I/O unit and are program-testable. These indicators are: Not Ready, Busy, Data Check, Condition, Wrong-Length Record, and No Transfer.

When the Op code is read out and the card punch is addressed, the I/O channel-status indicators are set with the status of the card punch. If the punch status is such that the punch command can be executed, no indicators are set and the transfer of data to punch storage occurs. Any indicator that is ON at this time prevents the data transfer. If an error occurs during the transfer, an indicator is set and the card is not punched.

In either case, any indicator that is set remains ON to indicate to the stored program that the card is not punched. For a more detailed description of the conditions that set the status indicators, refer to the IBM 1410 Data Processing System Reference Manual, A24-1407.

After the punch command is given, the status indicators for the I/O channel must be tested before that channel can be used again. The Op code for testing the status indicators on the channel that is associated with the card punch is RI  $\frac{1}{2}$ .

#### Operation (Figure 3.1-9)

Instruction read-out places the instruction word characters in various registers to control the execution of the card punch Op code.

At I-Op time, the Op code character is read out and placed into the Op register. If the character is an M, it specifies the move mode, and word marks in the data

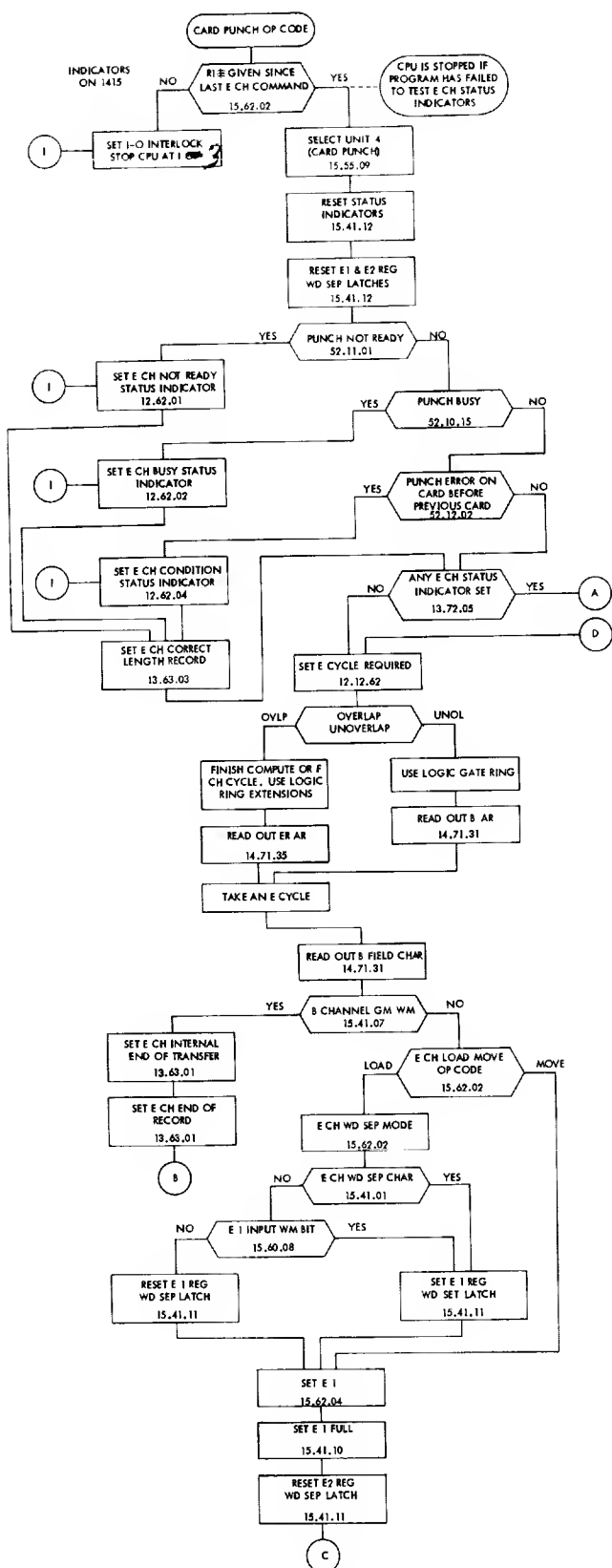


Figure 3.1-9A Punch Card Logic Flow

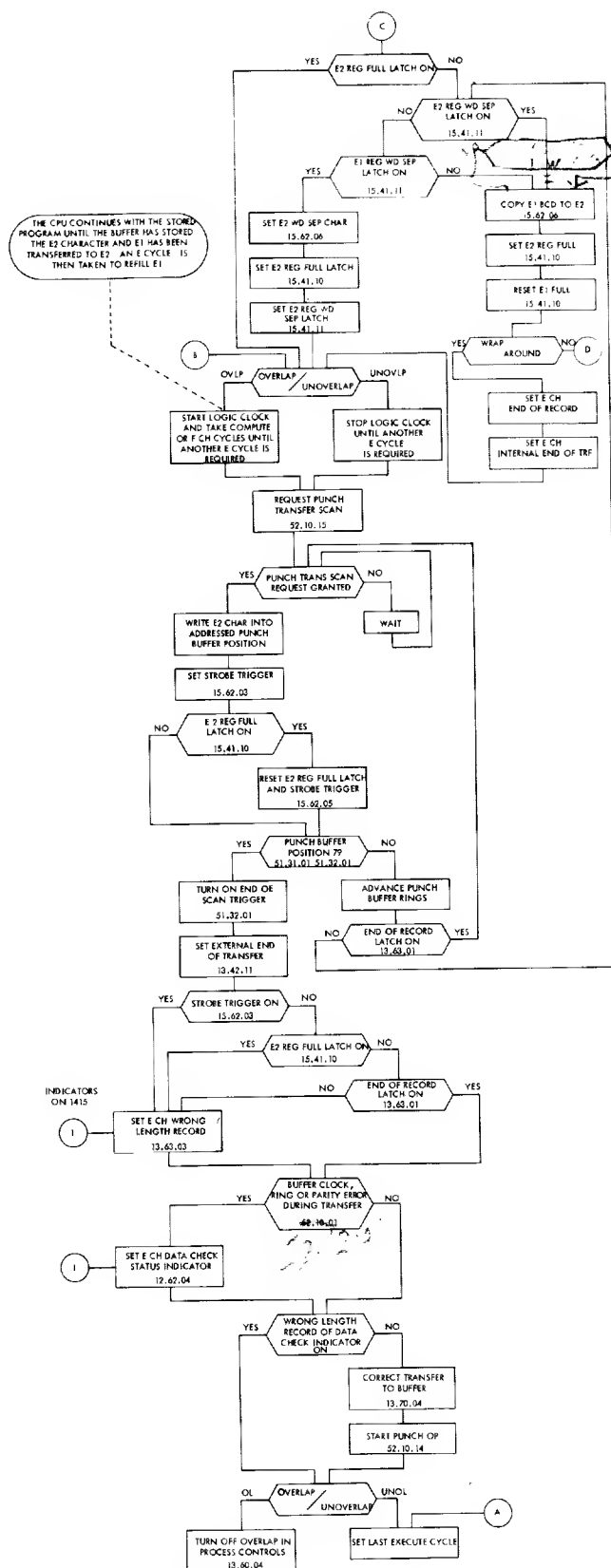


Figure 3.1-9B Punch Card Logic Flow



record are disregarded. An L Op code specifies the load mode so that word marks can be translated to word separators and sent to punch storage.

All data or control characters transfer to punch storage via the E-channel. The E-channel is one of two possible I/O channels available on the 1410 system. At I1-I3 time, the hundreds position of the X-control field reads out and is placed into the I/O channel-select register to specify an overlapped or unoverlapped operation. Because the card punch is assigned to the E-channel, this character can be either a % for unoverlap or a @ for overlap. Also at I3 time, an I/O interlock check is made to insure that the E-channel-status indicators have been tested by the program since the last E-channel I/O command. If the program failed to test the indicators, the I/O interlock results in a master error that stops the CPU.

The tens position of the X-control field is the unit-select character, and is a 4 so that the card punch is selected. Because the E-channel is selected at I3 time, the unit-select character (read out at I4 time) is placed in the E-channel unit-select register. The 4 in the unit-select register addresses the card punch and gates the punch status to the CPU, in preparation for setting the status indicators later in I phase.

At I5 time, the unit-number character is read out and placed into the E-channel-unit number register. At the same time, the character transfers over the E-channel to the punch-storage circuitry to select the stacker for the card at the check station. At I5 time, the unit-number character is set into the E1 data register. From the E1 data register, the stacker-select character transfers to the E2 data register where it becomes available to the punch-storage circuitry. Decoding the character in the punch-storage circuitry causes the proper stacker-select latch to set and subsequently the proper-stacker magnet to energize.

At I6 through I10, the address of the high-order position of the record to be punched reads out and is placed either in the B- and D-address registers for an unoverlapped command, or in the B-, D-, and E-address registers for an overlapped command. The B-address register is used to scan the storage output field on an unoverlapped transfer. The E-address register is used to scan the storage output field.

The Op modifier character reads out at I11 time and is set into the Op modifier register. For the punch Op code, the Op modifier character is a W to specify a write, or output operation.

The CPU normally takes another I-cycle to search for the word mark in the next position. During this last instruction read-out cycle, a status sample A-pulse develops to set the appropriate status indicators with the status of the punch. Because the punch is not ready, or busy; or because there is an error on the card before the previous card, the CPU sets the E-Channel Not Ready, E-Channel Busy, and E-Channel Condition status indicators, respectively. If any of these indicators are set at this time, the CPU does not execute the instruction. Instead, it reads out the next instruction in sequence. The punch-error circuit is reset following the status sample A-pulse, because a punch error is now stored in the E-channel-condition status indicator. If no indicators are set at status sample A-time, it means that the punch can execute the punch command.

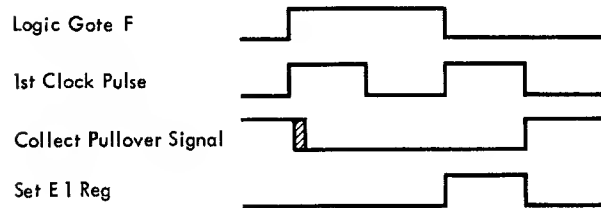
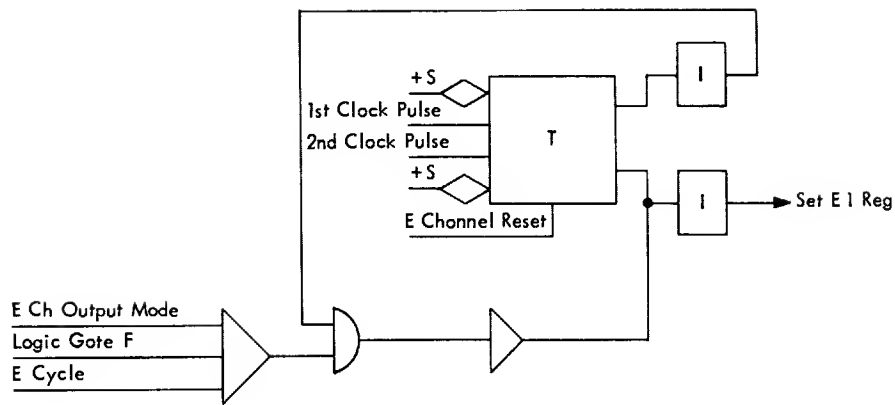


Figure 3.1-10 E1 Register Set Pulse

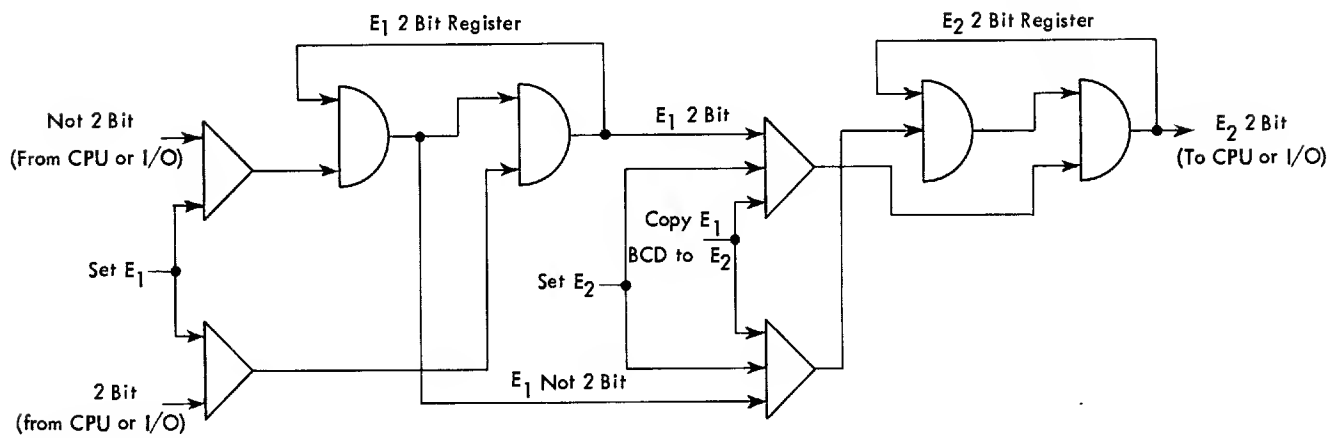


Figure 3.1-11 E-Channel Register For 2-Bit

The CPU then goes into execute phase. If the command is overlapped, the EAR is read out and set into the STAR. An E-cycle is taken to read out a character. The character is gated from the B-data register, through assembly, and to the input-output registers. The I/O channel that is used to transfer data to punch storage is terminated by two single-character registers called the E1 and E2 data registers. Two registers are necessary to synchronize the CPU and the I/O unit, and to allow word marks to translate to word separators, when the machine is in the load mode.

Once a character reads out and is gated to the input of the E1 register, a Set E1 Reg pulse develops to set the character into the E1 register. A trigger controlled by collector pull-over develops the required set pulse. This type of circuit is used frequently in I/O operations to allow a pulse from an I/O unit to control a CPU function.

For an output operation, the Set E1 Reg pulse occurs at the first clock pulse of logic gate G (Figure 3.1-10). The trigger is normally held OFF by collector pull-over, and the first and second clock pulses at the trigger inputs have no effect. Logic gate F and the output cycle are switched to remove the trigger collector pull-over. Due to circuit delays, the trigger cannot flip until it receives the first clock pulse of logic gate G. The trigger ON output feeds back to block the collector pull-over line, which would otherwise rise with the fall of logic gate A. When the second clock pulse flips the trigger OFF, the collector pull-over line holds the trigger in the off status. The resultant pulse is one clock pulse in length occurring at logic gate G.

The E-data registers for a 2-bit are shown on Figure 3.1-11. Notice that there is no direct resetting of these registers. Instead, each register is set by the presence of a bit at the input, and reset by the presence of the not-bit.

There are four controls to transfer data from the E1 to the E2 register. The functions of the transfer control are:

1. Copy E1 BCD to E2 Reg. This line gates the E1 BCD bits (B, A, 8, 4, 2, 1) to the input of the E2 register.
2. Set E2 Word Separator. This line generates a word-separator character (C, A, 8, 4, 1) in the E2 register.
3. Copy E1 WM · C-Bit. This control gates the E1 register WM and C-bits to the input of the E2 register.
4. Copy Inv WM · C-Bit. This line inverts the E1 register WM and C-bits, and gates them to the input of the E2 register.

If E2 is not full, it is filled (set) with the E1 data according to the controls that are set up. The transfer controls are set up by the word-separator control circuit. With E2 register full, a signal is sent to punch storage to request a transfer scan. When priority grants the scan request, punch storage begins taking memory cycles to scan the punch-storage unit, one position at a time. During each memory cycle, the character in the E2 register is set into the addressed position of punch storage, and a strobe pulse is sent back to the CPU to indicate that the character was stored.

This strobe causes the character in E1 to transfer to E2. The CPU is then signalled to take another E-cycle in order to place a new character into E1.

This sequence of events repeats until the punch-storage End of Scan trigger signals that all storage positions are scanned. The transfer terminates, and a status sample B-pulse develops to set the E-channel wrong-length record and data-check-status indicators, in case the transfer was incorrect. An error during the transfer prevents the punch clutch from energizing. If no errors occur during transfer, the punch clutch energizes, and the record is punched into the card. In either case (error or no-error), the CPU is released to continue processing.

## Circuits

Common Op-code grouping lines are:

Percent type op codes  
 Not addr dbl op codes  
 2 addr plus mod op codes  
 2 address op codes  
 No index on 1st addr ops  
 No branch op codes  
 M or L op codes

1. Gate the unit-number character to punch the stack-select latches.

<u>Signal</u>	<u>Control</u>	<u>Logic</u>
Sw B-ch to A- reg	I-cycle Logic gate D 2nd clock pulse 3	15. 38. 01
Use A-ch nu	I-cycle	15. 49. 03
Use A-ch zones	I-cycle (not) 1401 · I-ring 5 or 10 Time · I-cycle	15. 49. 03
Use A-ch WM	I-cycle	15. 49. 02
Set E1 reg	Logic gate A I-ring 5 time I-cycle 1 M or L op codes E-Ch sel unit 4 Percent or coml at	15. 62. 04
Set E2	E1 reg full (not) E2 reg full	15. 62. 04
Mod stk sel	(not) 1401 mode CPU to I/O trans Select unit 4 Strobe latch (off) Time 100-000	51. 40. 43

<u>Signal</u>	<u>Control</u>	<u>Logic</u>
Pch stk sel	Mod stk sel	51. 40. 50
Stack____latch (on)	Pch stk sel CPU to I/O bit____	52. 13. 01

2. Set E-Channel Not Ready, E-Channel Busy, and/or E-Channel. Condition status indicators according to the status of the card punch.

<u>Signal</u>	<u>Control</u>	<u>Logic</u>
Pch not ready to drive	(punch jam) (punch off line) (punch power off) (punch stop)	52. 11. 01
(not)pch ready	Pch not ready to drive	52. 11. 02
(not) buffer ready	(not) pch ready Select unit 4	51. 40. 18
E-ch sel any buffer	Select unit 4	13. 60. 03
(not) E-ch ready bus	E-ch sel any buffer (not) Buffer ready	12. 62. 01
E-ch status sample A	M or L op codes I-ring 12 time (not) E-ch select unit F I-cycle Percent or coml at (not) file op Logic gate D (not) 1401 mode	13. 65. 06
E-ch not ready	E-ch status sample A (not) E-ch ready bus	12. 62. 01
Pch busy	(Punch clutch energized)	51. 40. 19
Buffer busy	Pch busy Select unit 4	51. 40. 19
E-ch busy bus	Buffer busy E-ch sel any buffer	12. 62. 02
E-ch busy	E-ch busy bus E-ch ready bus E-ch status sample A	12. 62. 02
Pch error	(hole count error, previous card feed cycle) or (clock error, previous card feed cycle) or (ring error, previous card feed cycle) or (parity error, previous card feed cycle)	52. 12. 02

<u>Signal</u>	<u>Control</u>	<u>Logic</u>
Buffer conditions	Pch error Select unit 4	51. 40. 20
E-ch sel any buffer	Select unit 4	13. 60. 03
E-ch condition bus	Buffer condition E-ch sel any buffer	12. 62. 03
E-ch condition	E-ch condition bus E-ch status sample A (not) File op (not) E-ch select tape data E-ch ready bus (not) E-ch busy bus	12. 62. 04

3. If any indicator is set at status sample A-time, go on to the next instruction in sequence.

<u>Signal</u>	<u>Control</u>	<u>Logic</u>
E-ch correct length record	Set E-ch condition (or) Set E-ch busy (or) Set E-ch not ready	13. 63. 03
E-ch any status (on)	E-ch correct length record	13. 72. 05
Last execute cycle I/O	E-ch any status (on) I/O percent latch Last insn RO cycle (not) File op	13. 65. 07
Last execute cycle	Last execute cycle I/O	12. 12. 51

4. If the condition latch is on, reset the punch error.

<u>Signal</u>	<u>Control</u>	<u>Logic</u>
E-ch status sample A-delay	Logic gate E B-ch WM M or L op codes I-ring 12 time (not) E-ch select unit F Percent or coml at I-cycle	13. 65. 06
Reset select buffer latches	E-ch condition E-ch select any buffer E-ch status sample A-delay	13. 70. 04
Reset CPU	Reset select buffer latches	51. 40. 06
Pch error reset	Reset CPU Select unit 4	52. 12. 01
Pch check 2 (off)	Pch error reset	52. 12. 02

5. If no status indicators are on, request an E-cycle.

<u>Signal</u>	<u>Control</u>	<u>Logic</u>
E-cycle required	E-ch output mode (not) E-ch int end-of-transfer (not) E1 reg full (not) E-cycle any last gate E-ch in process	12. 12. 62
E-cycle control	E-cycle required 2nd clock pulse Logic gate Z (not) E-ch int end-of-transfer (not) F-cycle required	12. 12. 66

6. Start the appropriate logic clock and transfer the contents of the proper address register to STAR.

- a. If the instruction is unoverlapped, start the normal logic clock (A-K) and read out BAR.

<u>Signal</u>	<u>Control</u>	<u>Logic</u>
(Start logic clock)	E-cycle required E-ch unovlp-in-process	11. 10. 10
RO BAR	E-cy unovlp-in-process E-cycle ctrl (not) Console inhibit AR RO (not) 1st I/O cycle control (not) E-ch 2nd addr unovlp	14. 71. 31

- b. If the instruction is overlapped, start the logic-clock extension ring (R-Y) and read out EAR.

<u>Signal</u>	<u>Control</u>	<u>Logic</u>
(Start logic-extension clock)	E-cycle required E-ch overlap-in-process	11. 10. 20
RO EAR	E-cycle ctrl E-ch ovlp-in-process (not) Console inhibit AR RO	14. 71. 35

7. Take an E-cycle to read out the addressed character.

<u>Signal</u>	<u>Control</u>	<u>Logic</u>
E-Cycle	E-cycle ctrl Logic gate B or S	12. 12. 66

8. Gate B-channel character to E1 input.

<u>Signal</u>	<u>Control</u>	<u>Logic</u>
Output cycle	E-ch output mode E-cycle	13. 60. 06
Output-field cycle	E-cycle (not) E-ch select unit 2	15. 49. 04
Use B-ch nu	Output cycle Output-field cycle	15. 49. 05
E-ch select odd parity unit	E-ch select unit 4	13. 60. 03
Odd parity cycle	E-cycle E-ch select odd parity unit	13. 60. 02
Use B-ch zones	Output cycle Output-field cycle Odd parity cycle	15. 49. 05
Gate Asm ch to E1 input	E-ch output mode (not) Control reg disable (not) 1401 B-op code	15. 62. 07

9. If the B-channel character is a group-mark with a word-mark, end the internal transfer.

<u>Signal</u>	<u>Control</u>	<u>Logic</u>
B-ch group mark · WM	B-ch 1 · 2 · 4 · 8 · A · B · WM	15. 41. 07
E-ch end-of-record latch	E-cycle B-cy GM · WM Logic gate F W symbol op modifier (not) 1401 cd-print-in-process or E-cycle B-ch GM WM Logic gate W	13. 63. 01
E-ch int end-of-transfer	(same as E-ch end-of-record)	13. 63. 01

10. Set E1 data register. If the instruction is given in the load mode, analyze the E1 input for a word mark or word-separator character, and set up controls for E1 to E2 transfer.

<u>Signal</u>	<u>Control</u>	<u>Logic</u>
Set E1	E-cycle E-ch output mode (not) E-ch 1st char 2nd addr B-ch not group mark · WM Logic gate F or W	15. 62. 04
E-ch select 7-bit unit	E-ch select unit 4	13. 60. 03



<u>Signal</u>	<u>Control</u>	<u>Logic</u>
E-ch word-separator mode	E-ch select 7-bit unit E-ch load mode	15. 62. 02
E-ch word separator	E1 input $C \cdot B \cdot 4 \cdot 2 \cdot 1 \cdot A \cdot B$ -bits	15. 41. 01
E1 input WM bit	Assembly ch WM bit Gate asm ch to E1 input	15. 60. 08
E1 reg word-separator latch(on)	E-ch word separator E-ch word separator mode Set E1 or E1 input WM bit E-ch word-separator mode Set E1	15. 41. 11
E1 reg not word separator	(same as E1 reg word separator latch on)	15. 41. 11
E1 reg full	Set E1	15. 41. 10
E2 reg word separator latch(off)	E-ch output mode Set E1 reg	15. 41. 11
E2 reg not word separator latch(on)	E-ch output mode Set E1 reg	15. 41. 12

11. Transfer E1 to E2. If the instruction is given in a move mode, set E2 with E1 BCD.

<u>Signal</u>	<u>Control</u>	<u>Logic</u>
E-ch reset	I-ring 4 time Percent or com1 at Logic gate C	15. 41. 12
E1 reg not word separator	E-ch reset	15. 41. 11
E2 reg full latch (off)	E-ch reset	15. 41. 10
Set E2 reg	E1 reg full latch (on) E2 reg full latch (off)	15. 62. 04
E2 reg full latch (on)	E-ch output mode Set E2 reg	15. 41. 10
E1 reg full latch (off)	E1 reg not word separator Set E2 reg	15. 41. 10

Notes: When the E1 reg full latch is turned OFF, the E-cycle-required line is brought up to cause another character to read out and to be gated to the E1 register.

12. Transfer E1 to E2. If the instruction is given in the load mode, and the E1 reg not word separator latch is ON, copy E1 BCD to E2.

<u>Signal</u>	<u>Control</u>	<u>Logic</u>
E1 reg not word separator	E-ch not word separator (not) E1 WM bit Set E1 reg	1S. 41. 11
Copy E1 BCD to E2	E1 reg not word separator	1S. 62. 06
E2 reg full latch (on)	E-ch output mode Set E2 reg	1S. 41. 10
E1 reg full latch (off)	E1 reg not word separator Set E2 reg	1S. 41. 10

13. Transfer E1 to E2. If the instruction is given in the load mode, if E2 reg not word-separator latch is on, and if E1 reg word-separator latch is on, set the E2 reg word-separator character.

<u>Signal</u>	<u>Control</u>	<u>Logic</u>
Set E2 word separator	Set E2 reg (not) Gate console to assembly E1 reg word separator (not) E2 reg word separator	1S. 62. 06
Set E2 reg delayed	Set E2 reg	1S. 62. 04

14. Transfer E1 to E2. If the instruction is given in the load mode and E2 reg word-separator latch is on, copy E1 BCD to E2.

<u>Signal</u>	<u>Control</u>	<u>Logic</u>
Copy E1 BCD to E2	E2 reg word-separator latch(on)	1S. 62. 06

15. Request a punch-transfer scan.

<u>Signal</u>	<u>Control</u>	<u>Logic</u>
Ready to buffer	E-ch-in-process E-ch select any buffer E2 reg full	13. 70. 04
Ready	Ready to buffer	S1. 40. 04
AC set pch trans reg trig	Pch ready Ready (not) Pch busy	S2. 10. 14
Pch trans req	Select unit 4 AC set pch trans req trig	S2. 10. 15

16. Write the E2 character into addressed punch-storage position.

<u>Signal</u>	<u>Control</u>	<u>Logic</u>
CPU to I/O sync___bit	E2 reg___bit	15. 60. 31-34
CPU to I/O bit___	CPU to I/O sync___bit	51. 40. 10
Inh gate___bit	CPU to I/O bit___	51. 16. 03
Write pulse	070-100	51. 30. 05

17. Set the strobe trigger.

<u>Signal</u>	<u>Control</u>	<u>Logic</u>
Trans scan	Pch trans scan	51. 40. 40
Strobe latch	Trans scan Time 020-030	51. 40. 43
CPU to I/O trans	Pch trans scan	51. 40. 40
Buffer strobe	Strobe latch CPU to I/O trans Time 100-000	51. 40. 43
E-ch strobe trigger	Buffer strobe E-ch select any buffer	15. 62. 03
Reset E2 full latch	E2 reg full E-ch strobe trigger E-ch output mode	15. 62. 65
E2 reg full latch (off)	Reset E2 full latch	15. 41. 10
E-ch strobe trigger (off)	Reset E2 full latch E-ch output mode	15. 62. 03

18. After 80 memory cycles, stop scanning the punch storage unit, and end the external transfer.

<u>Signal</u>	<u>Control</u>	<u>Logic</u>
End-of-scan	Tens-ring AC Set Tens-ring 7	51. 32. 01
End-of-transfer 1 latch (on)	Trans scan	51. 40. 12
End-of-transfer 2 latch (on)	End-of-transfer 1 latch (on) Trans scan (off) Time 090-100	51. 40. 12
End-of-transfer	End-of-transfer 2 latch (on)	51. 40. 12

19. Test for wrong-length record and/or for data check. Set the appropriate E-channel status indicators.

<u>Signal</u>	<u>Control</u>	<u>Logic</u>
E-ch int end-of-trf delayed	E-ch int end-of-transfer Logic gate Z	13. 65. 05
E-ch status sample B	E-ch int end-of-trf delayed Logic gate Z E-ch ext end-of-transfer (not) 2nd clock pulse	13. 65. 05
E-ch wrong-length record	E-ch status sample B E2 reg full or (not) E-ch end-of-record latch or E-ch strobe trigger	13. 63. 03
E-ch correct-length record	E-ch status sample B E-ch end-of-record latch (not) E1 reg full (not) E2 reg full (not) E-ch strobe trigger	13. 63. 03
Error	Pwr clock error or Parity or Ring error to error latch	52. 12. 01
Pch transfer error	Error Pch scan Pch trans req	52. 12. 01
Buffer error	Pch transfer error Select unit 4	51. 40. 21
E-ch check bus	Buffer error E-ch select any buffer	12. 62. 03
E-ch check	E-ch check bus E-ch status sample B	12. 62. 04

20. If instruction is unoverlapped, initiate the last execute cycle.

<u>Signal</u>	<u>Control</u>	<u>Logic</u>
Last execute cycle I/O	E-ch unovlp-in-process (not) 1401 card or print op code E-ch status sample B	13. 65. 07
Last execute cycle	Last execute cycle I/O	12. 12. 51

21. If no indicators are set, start the punch operation.

<u>Signal</u>	<u>Control</u>	<u>Logic</u>
E-ch second sample B	E-ch status sample B (not) 2nd clock pulse	13. 65. 05
Correct trans to buffer	E-ch correct-length record (not) E-ch check E-ch select any buffer E-ch second sample B	13. 70. 04
Go	Correct trans to buffer	51. 40. 04
Punch feed gate	Go Select unit 4	52. 10. 14

22. Reset the punch-transfer error latch (if on).

<u>Signal</u>	<u>Control</u>	<u>Logic</u>
Reset select buffer latches	E-ch sel any buffer E-ch second sample B	13. 70. 04
Reset CPU	Reset select buffer latches	51. 40. 06
Pch transfer error (off)	Reset CPU Select unit 4	52. 12. 01

23. Reset the overlap or unoverlap-in-process latches.

<u>Signal</u>	<u>Control</u>	<u>Logic</u>
E-ch status sample B-delay	E-ch second sample B (not) 2nd clock pulse	13. 65. 05
E-ch overlap-in- process (off)	E-ch status sample B-delay	13. 60. 04
E-ch unoverlap-in- process (off)	E-ch status sample B-delay	13. 60. 04

### 3. 2. 00 CARD READER

When the 1402 reader start key is pressed, the reader clutch energizes and cards feed into the read feed. As the first card passes under the first set of read brushes, the card is read and the information transfers to the hole-count check circuits. At the next set of read brushes, the card is read again. This time, the information is encoded to BCD form, and is placed in the read core-storage unit; in addition, it is sent to the hole-count check circuits (Figure 3. 2-1). When the cards are run in, and when the first card in is completely checked, the information in reader storage can be transferred to the CPU.

To execute the card read command, the CPU transfers the record from reader storage to some specified CPU core-storage location. When the transfer is complete, the reader clutch can be energized to cause the next card to read in and be checked.

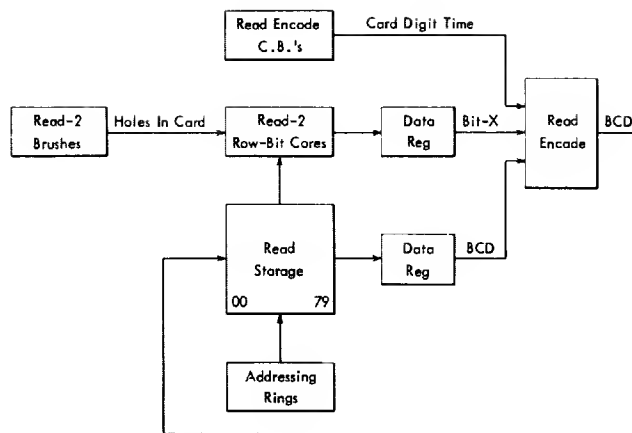


Figure 3.2-1 Card Read-In Data Flow

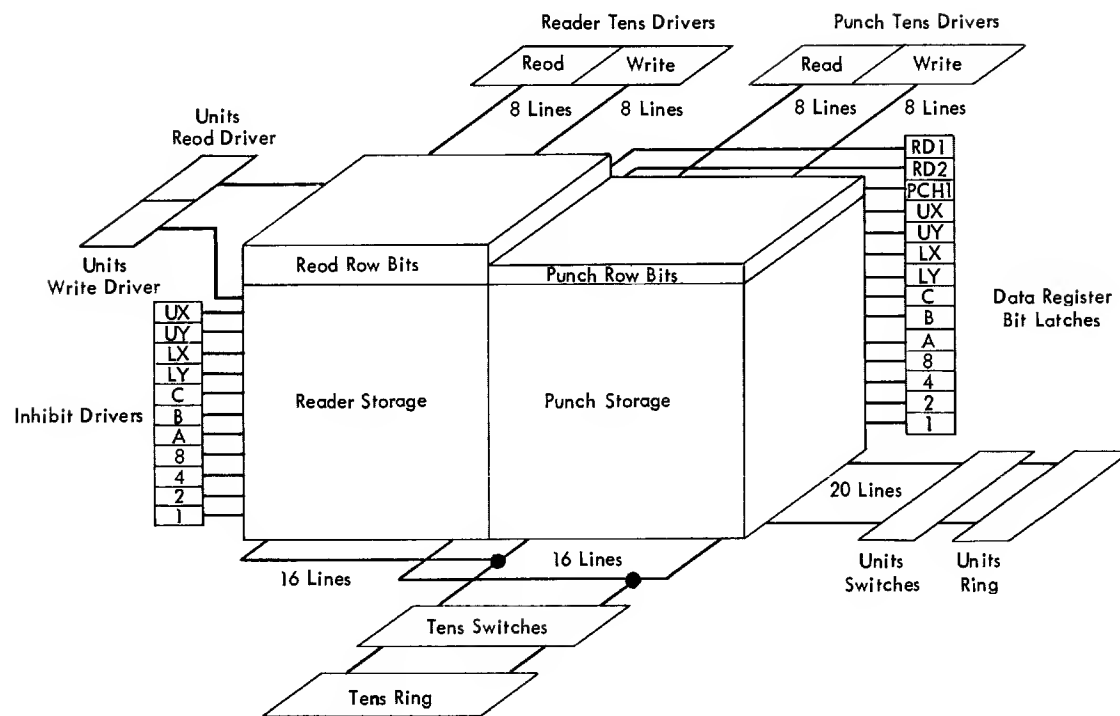


Figure 3.2-2 Common Addressing Scheme

### 3.2.01 Core Storage

The reader core-storage unit is similar to the punch-core storage unit. Reader storage has one additional row-bit core plane, because the reader has two sets of read brushes whereas the card punch has only one set.

The units and tens rings, units and tens switches, and units drivers that address punch storage also address reader storage (Figure 3.2-2). The units ring and switches select one of ten rows of cores in one direction, while the tens ring and switches are selecting one of eight rows of cores in the other direction. The coincidence of the two lines selects the same relative position in each storage unit (reader-punch). The units-read, or the units-write driver, is impulsed to cause current to flow through either the units-read or units-write winding. By impulsing the proper tens driver (read or write, reader or punch) the desired core is either read into or read out.

The read windings pass through all 13 planes. This means that each time a position is addressed to read out, all 13 cores for that position read out. The write windings pass through all planes except the row-bit planes. The reader row-bit planes are set by the coincidence of a half-write winding, and a six-turn winding that is connected to each read brush.

### 3.2.02 Read-Storage Scanning and Priority Controls

Reader storage is scanned the same as punch storage. That is, once a scan starts, the units and tens rings advance so that the 80 positions are addressed in sequence.

A scan of reader storage is requested after a card row reads into the row-bit storage. When it is desired, a scan can be requested in order to transfer the reader storage contents to the CPU (Figure 3.2-3). Two priority conditions must be met before a scan request can be granted.

1. There must be no scan in progress.
2. There must be no higher priority scan request.

The impulses from the reader circuit breakers (CB's) signal that a read scan is to be requested. A read transfer scan is requested by the CPU. Either condition develops a read priority gate (Figure 3.2-4). Read scan timing is shown in Figure 3.2-5.

The read scan request is never initiated until all higher-priority scan requests are satisfied. Under certain conditions, it is possible for the reader impulse CB's to make and to develop a brush impulse before the previous read scan is complete. Under these conditions, if the brushes were to be driven as soon as the brush impulse CB's make, new row-bit data would read in on top of the old.

To prevent this overlapping of the scan and of the succeeding brush impulse, the brush impulse CB switches through the priority controls to insure that preceding scan is complete before it supplies another brush impulse (Figure 3.2-4). Note that the read request trigger must be OFF due to an end-of-read-scan condition before the brush single shot can be impulsed. This produces a sliding brush impulse

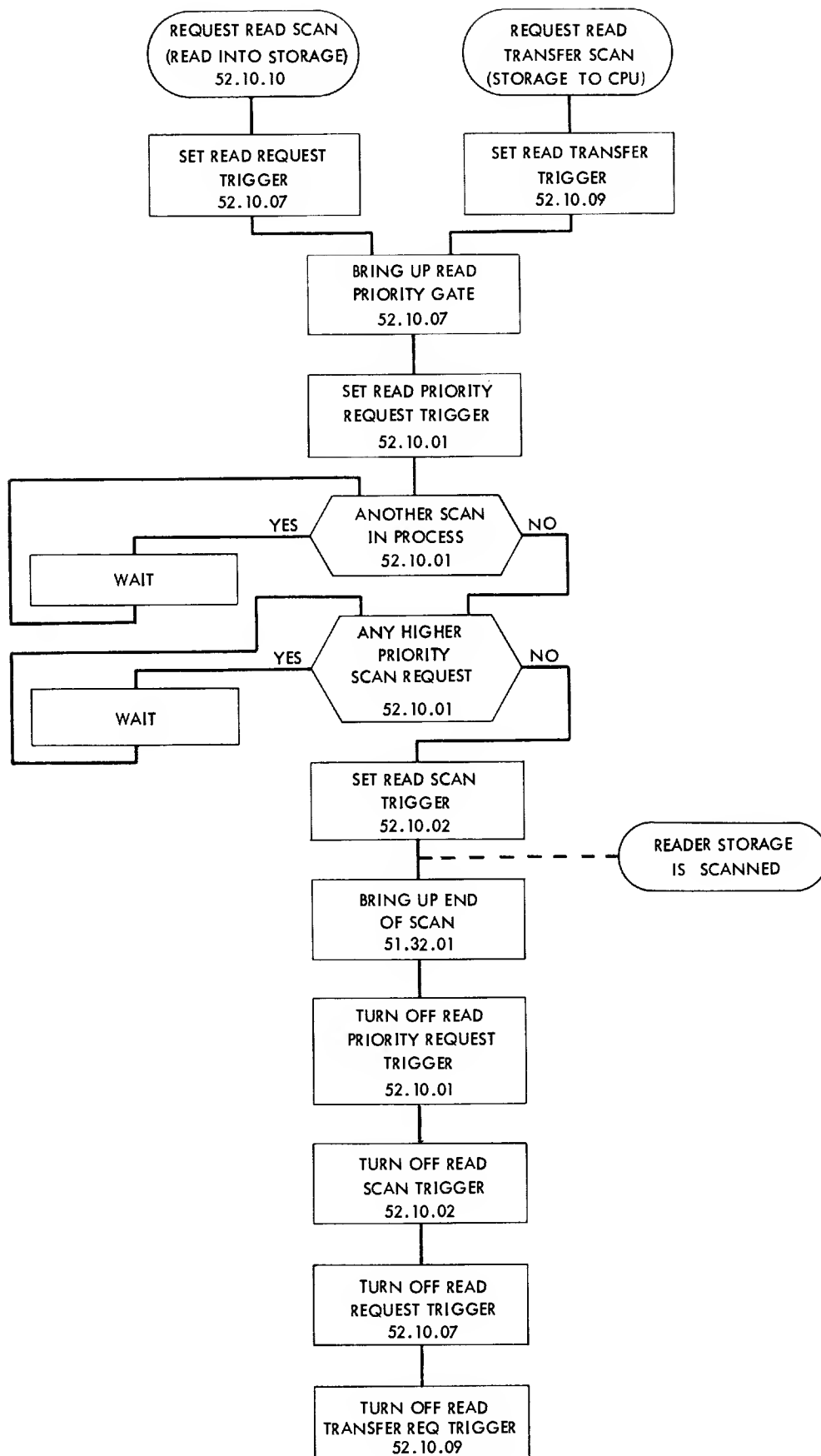


Figure 3.2-3 Card Reader Priority Controls



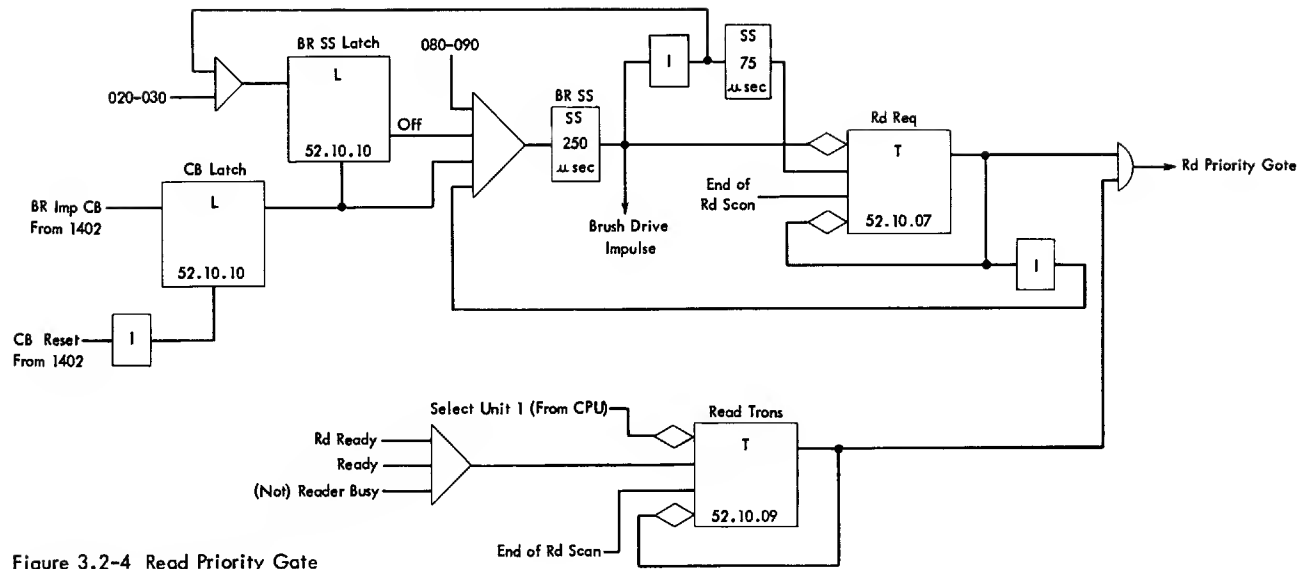
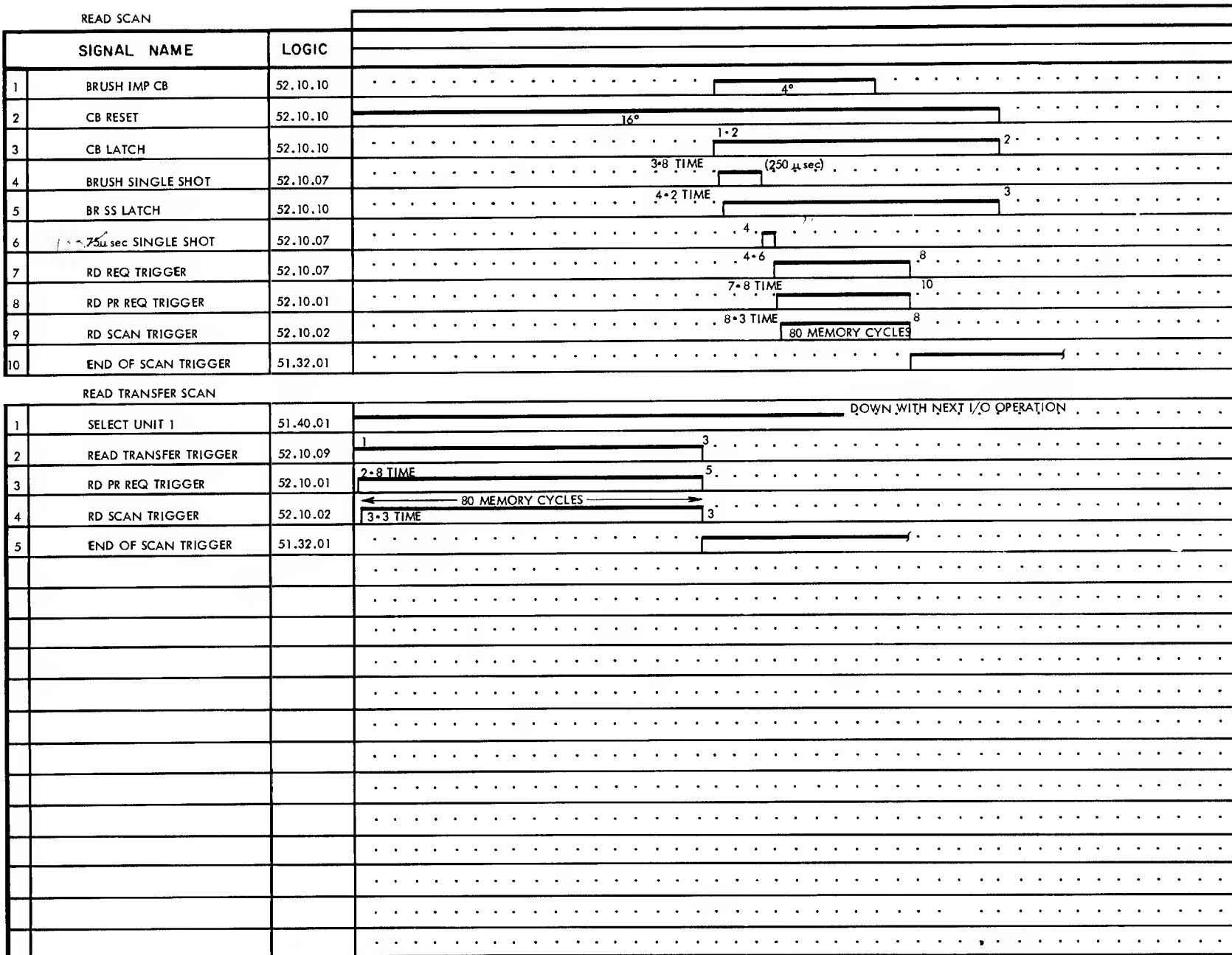


Figure 3.2-4 Read Priority Gate

### Figure 3.2-5 Read-Scans Timing Chart



that is delayed, if necessary, until the previous read scan is complete. The brush impulse is developed by the 250  $\mu$ s single shot. The 75  $\mu$ s single shot delays the start of the scan until the brush-drive transistor has time to turn off.

### 3.2.03 Read Encode

Information that is read from Read Two-Row Bit-Storage feeds to the read encoder where it converts from card-row form to BCD form. This BCD information reads into the data planes of read storage. At the end of the card feed cycle, it becomes available to the CPU. Because of alphametical or special characters, more than one hole can be read in any column. Therefore, provision must be made to combine the bits of all punched holes into a valid character. As each card column of each cycle point is coded, it is combined with data bits already in that position of storage from a previous cycle point. The result is a bit representation of all punches that are read in a particular column.

Information from the Read Two-Row Bit-Storage data register can represent any character. This information is referred to as Bit X information. An encode signal that is developed by a read request and read scan, combines with Bit X to produce an Encode Bit X signal. This signal activates the read encode circuits.

The read encode process takes place on every memory cycle; that is, 80 times for each read scan. This allows the machine to encode every one of the 960 possible punched-holes in an IBM card.

The read encode circuits are on Logics 51.12.02; 51.12.03; 51.12.04. Refer to these pages, and to the Reader CB Timing Chart (1402-2 Wiring Diagram). A discussion of the development of a representative character follows.

Assume that a 3 punch is read in the card. At reader 9 time, a C-bit is inserted (Logic 51.12.04). This is necessary, because an odd-bit count must be maintained for every character in storage, even if the character is not completed. On the corresponding memory cycle of each successive read scan through 4, this C-bit is regenerated. At reader 3 time, CB1 and CB2 combine with Bit X (Logic 51.12.03) to produce Read Encode Out 1-Bit and Read Encode Out 2-Bit. Notice that CB-C is made whenever two bits are encoded simultaneously, because the addition of an even number of bits requires that the C-bit be reinserted.

At 2-through-12 time of the reader, no more holes are read. Therefore, the C-bit is reinserted at each cycle point. At the end of read scan 12 (the end of the reading of the card), reader storage contains a 2-bit, a 1-bit, and a C-bit. This is the correct BCD coding for the 3 punch in a card.

### 3.2.04 Stacker Select and Feed Op Code

#### Op Code Function (Od)

The K Op code causes the card reader to take a card-feed cycle, to read and to check the cards that are passing under the read brushes, and to select the previous card into one of the read stackers.

## Op Modifier Function

The 1, 2, and 8-bits of the Op modifier character determine which read stacker is selected. A 1-bit selects the 1-stacker; a 2-bit selects the 2-stacker; and an 8 and 2-bit combination selects the normal read (N/R) stacker.

### Operation (Figure 3.2-6)

At I-Op time, the K reads out and is placed in the Op register where it is decoded in order to bring up the common Op-code grouping lines. If the E-channel status indicators have not been program tested since the last I/O command on the E-channel, an I/O interlock check stops the CPU.

The K-Op code forces a 1 in the E-channel-unit select register to address the card reader. Select unit 1 gates reader status to the E-channel-status indicators. The Op modifier character that is read out at I1 time is gated to the A-channel, through assembly, to the E-data registers, and out to the reader-storage circuitry.

Status sample A occurs at I2 (last instruction-read-out cycle) to set the status indicators according to the reader status. If the reader is not ready, the E-channel not ready latch sets. If the reader is busy, the E-channel busy latch sets. Failure to transfer the reader-storage contents since the last feed command, sets the E-channel no transfer latch. If any indicators are set at this time, the command is not executed. However, if no indicators are set, the stack-select character is gated to the proper stack-select latch. The reader clutch then energizes, and the stack-select latch energizes the proper select magnet. With the read clutch energized, the impulse CB's RL-5, RL-6, RL-7, and RL-8, request a read scan for each of the twelve rows in the card. The twelve read scans transfer and encode the 80 columns of card information to read storage.

### Circuits (Figure 3.2-7)

Common Op-code grouping lines are:

No branch  
Op mod to A-ch on B-cycle ops  
Regen mem on B-cycle ops  
Not-address double  
2 char only op codes

1. Read out the Op code, and develop the stacker-select signal.

<u>Signal</u>	<u>Control</u>	<u>Logic</u>
Set Op reg	Logic gate E B ch WM bit 2 I-ring op time I-cycle	12. 13. 04

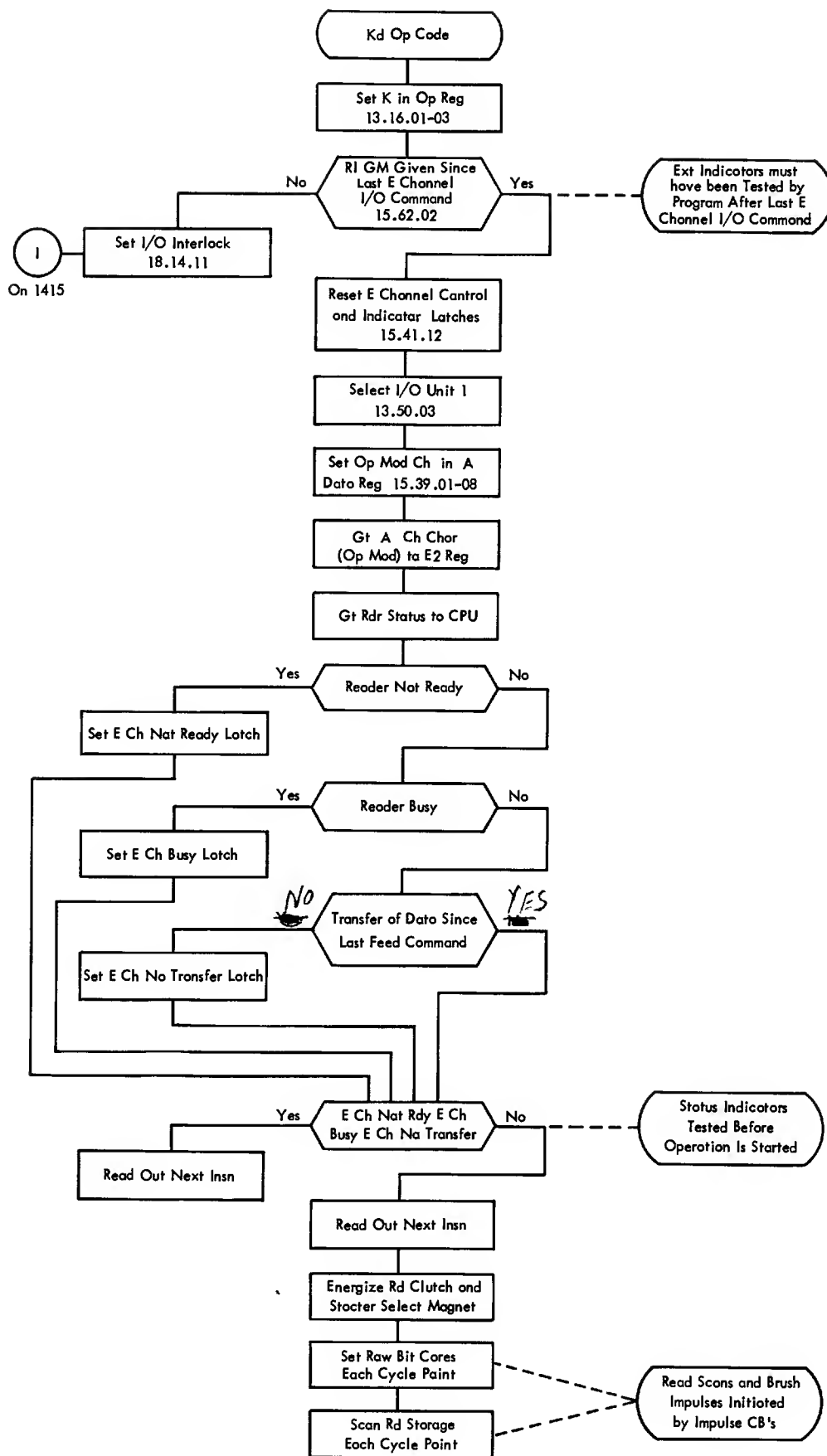


Figure 3.2-6 Kd Op Code

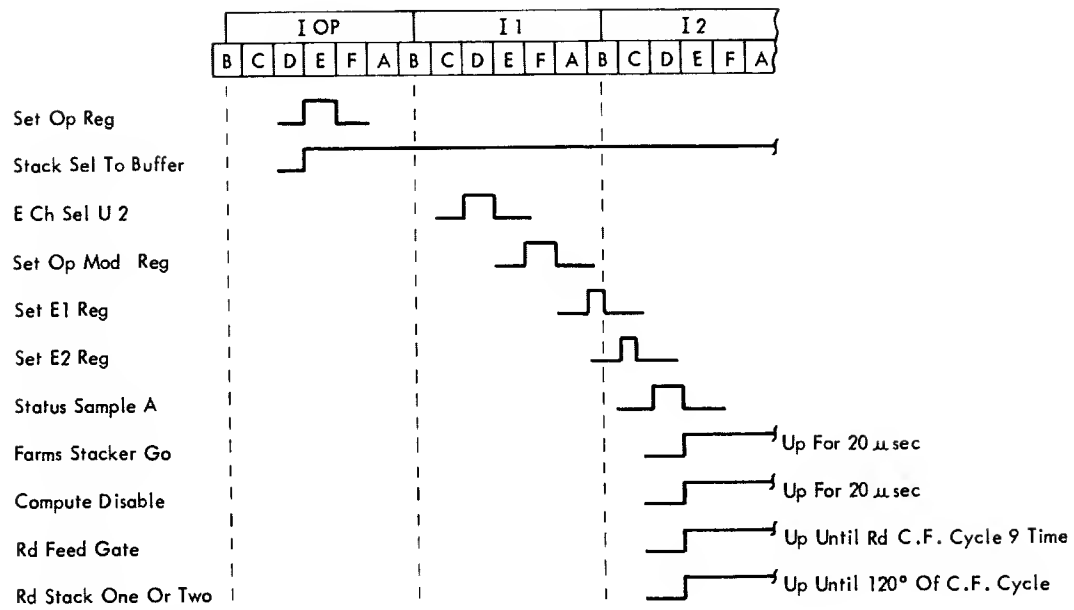


Figure 3.2-7 Kd Timing

<u>Signal</u>	<u>Control</u>	<u>Logic</u>
Stacker sel op code	Op dcd B not A not 8B Op reg com C-bit Op dcd not 4 · 2 not 1B	13. 13. 08
Stack select to buffer	Stacker sel op code	13. 70. 04

2. Test for I/O interlock (move and load latches should have been reset by RI $\ddagger$  command).

<u>Signal</u>	<u>Control</u>	<u>Logic</u>
(not) E-ch interlock	(E-ch move, load latches not set)	15. 62. 02
(not) I/O interlock check	(not) E-ch interlock	18. 14. 11

3. Reset the E-channel control and the status-indicator latches.

<u>Signal</u>	<u>Control</u>	<u>Logic</u>
F or K E-ch reset	Logic gate C I-ring 1 time 2 char only op codes	15. 41. 12
E-ch reset	F or K E-ch reset	15. 41. 12

4. Select unit 1 (card reader).

<u>Signal</u>	<u>Control</u>	<u>Logic</u>
E-ch U sel reg 1-bit	I-ring 1 + 1401 and 3 time Logic gate D Stacker sel op code	15. 55. 09
E-ch sel unit 1	E-ch U sel reg 1-bit E-ch U sel reg not (C · B · A · 8 · 4 · 2) bits	13. 50. 03

5. Set the Op modifier character in the A-data register.

<u>Signal</u>	<u>Control</u>	<u>Logic</u>
Sw B-ch to A-reg	I-cycle Logic gate D 2nd clock pulse	15. 38. 01

6. Set the E-move mode.

<u>Signal</u>	<u>Control</u>	<u>Logic</u>
E-ch move mode	Last insn RO cycle 2 char only op codes Logic gate E	15. 62. 02

7. Gate A-channel character (Op modifier) to E2 register.

<u>Signal</u>	<u>Control</u>	<u>Logic</u>
A-ch bit	Sw B-ch to A-reg	15. 39. 01-08
Use A-ch num	I-cycle (not) 1401 I-ring 5 + 10 time · I-cycle	15. 49. 03
Use A-ch zones	I-cycle (not) 1401 I-ring 5 + 10 time · I-cycle	15. 49. 03
Asm ch__bit	Use A-ch num Use A-ch zones	15. 50. 01-10
Gate asm ch to E1 input	I-cycle not CR disable (not) E-ch-in-process (not) 1401 B-op code	15. 62. 07
Set E1 reg	2 char only op codes A-cy not WM bit Logic gate A I-cycle	15. 62. 04
E1 reg full	Set E1 reg	15. 41. 10
(not) E1 reg word separator	E-ch reset	15. 41. 11
Copy E1 BCD to D2	(not) E1 reg word separator	15. 62. 06
Set E2 reg	E1 reg full (not) E2 reg full	15. 62. 04
E-ch__bit	Set E2 reg	15. 60. 31-34

#### 8. Gate the reader status to the CPU.

<u>Signal</u>	<u>Control</u>	<u>Logic</u>
Reader busy	(Reader in use)	52. 10. 08
Buffer busy	Reader busy Select unit 1	51. 40. 19
Buffer ready	Rd ready to CPU Select unit 1	51. 40. 18
Buffer no transfer cond	Multi Rd Feed Latch (on) Stacker select	52. 10. 08

#### 9. Set the E-channel status indicators according to unit 1 (card reader) conditions.

<u>Signal</u>	<u>Control</u>	<u>Logic</u>
E-ch status sample A	(not) File op I-cycle I-ring 2 time 2 char only op codes Logic gate D	13. 65. 06



<u>Signal</u>	<u>Control</u>	<u>Logic</u>
E-ch not ready	(not) Buffer ready E-ch status sample A	12. 62. 01
E-ch busy	Buffer busy E-ch status sample A	12. 62. 02
E-ch no transfer latch	Buffer no-transfer cond E-ch status sample A	13. 72. 04

10. Test the status indicators and develop the forms stacker go signal.

<u>Signal</u>	<u>Control</u>	<u>Logic</u>
Forms stacker	(not) E-ch not ready (not) E-ch busy (not) E-ch no-transfer 2 char only op codes Last insn RO cycle Logic gate E	13. 70. 04

11. Select the stacker magnet according to the E-channel (Op modifier) character.

<u>Signal</u>	<u>Control</u>	<u>Logic</u>
Rd stk sel	Multi rd fd latch (off) (not) 1401 mode Forms stacker go Stacker select	52. 10. 08
Rd stk one (or) two latch	CPU to I/O bit (1 or 2) Rd stk sel	52. 13. 02

12. Energize the stacker select magnet.

<u>Signal</u>	<u>Control</u>	<u>Logic</u>
Stack 1 (or) stack 2	Rd stack one (or) two latch Read CB RC6	52. 13. 02

13. Energize the read clutch magnet.

<u>Signal</u>	<u>Control</u>	<u>Logic</u>
Rd feed gate	Rd stk sel (not) Multi rd feed latch	52. 10. 08
Rd clutch	Rd feed gate Read CB RC5	52. 10. 09

14. Develop the brush impulse for each card row.

<u>Signal</u>	<u>Control</u>	<u>Logic</u>
CB latch	Brush imp CB CB reset	52. 10. 10
Brush single shot	CB latch (not) BR SS latch	52. 10. 07

15. Initiate a read scan for each card row.

<u>Signal</u>	<u>Control</u>	<u>Logic</u>
75 $\mu$ s single shot	Brush single shot (timed out)	52. 10. 07
Rd priority gate	Brush single shot (timed out) 75 $\mu$ s single shot (timed out)	52. 10. 07
Gate on rd scan tr	Rd priority gate (not) Some scan (not) Pt rd scan	52. 10. 01
Read scan	Gate on rd scan tr	52. 10. 02

16. Advance units and tens rings to scan read storage.

<u>Signal</u>	<u>Control</u>	<u>Logic</u>
Time pl 1 latch (on)	Read scan	51. 30. 02
Data reg reset 2	Time pl 1 latch (on)	51. 30. 02
Units ring drive	Data reg reset 2	51. 31. 02
T-ring AC latch (on)	Units ring 9	51. 32. 02
Tens ring AC set	T-ring AC latch (on) Data reg reset 2	51. 32. 02

### 3. 2. 05 Reader Move or Load

#### Op Code Function (OXXXBBBBBd)

A reader move or load Op code causes the contents of reader storage in the integrated synchronizer to be transferred to the core-storage unit in the CPU. A group-mark with a word-mark must appear in the core-storage position immediately to the right of the location of the data record in the CPU. After the transfer, the reader can be set in motion to read the next card into reader storage.

The instruction can be executed in either a move or load fashion. When the M Op code specifies a move mode, the record in read storage transfers "as is" to the CPU. That is, word separators are transferred as word separators. If a load command is executed, a single word-separator character that is read from the card causes a word mark to be placed over the next character to be read in. Two successive word-separator characters from the read storage cause a single word-separator character to be placed in core storage in the CPU.

An overlapped or unoverlapped operation is determined by the character in the hundreds position of the X-control field. Because the card reader is assigned to the E-channel, this character can be either a % for unoverlap or @ for overlap. The tens position of the X-control field is the unit-select character and must be a 1 in order to select the card reader. The unit-number character (units position of the X-control field) is sent to the read-storage circuitry during instruction read-out in order to control stacking and feeding during the next card feed cycle. If the character is either 0, 1, or 2, the card is selected to the N/R, 1, or 8/2 stackers, re-

spectively. If the unit-number character is a 9, the card reader is not set in motion after the transfer is made. The B-address specifies the high-order position of the data record, because the record is reverse-scanned into the core-storage unit. The d-modifier character must be an R to indicate a read (or input) operation.

Operation (Figure 3.2-8)

Instruction read-out causes the instruction word characters to be placed in various registers to control the execution of the card read Op code.

At I-Op time, the Op-code character reads out and is placed into the Op register. If the character is an M, the move mode is specified, and word-separator characters in reader storage move to CPU storage as word-separator characters. An L Op code specifies the load mode to allow word marks to be generated in CPU storage from the reader-storage word-separator characters.

At I1 time, the I/O channel-select character reads out. The I-ring advances to I3 and the channel select character is placed into the I/O channel-select register. The I/O interlock check makes at I3 in order to insure that the E-channel status indicators are tested by the stored program since the last I/O command occurred that used the E-channel. The check consists of testing the move or load latch. Because the RI  $\bar{z}$  resets these latches, they should both be OFF at I3. Either latch ON at I3 results in an I/O interlock that produces a master error that, in turn, stops the CPU.

At I4, the unit-select character reads out and is placed in the E-channel unit-select register, because the E-channel is selected at I3 time. Unit-select character 1 gates the reader status to the CPU. An E-channel reset develops at I4 to reset all control and status-indicator latches that are associated with this channel.

The unit-number character reads out and is placed into the E-channel unit-number register at I5 time. This character is also gated to the E2 data register. The E2 full latch resets, but the character remains in the E2 register where it is available later for controlling the read feed.

At I6 through I10, the B-address reads out and is placed in the B- and D-address registers. In addition, the B-address is placed into the E-address register on an overlapped command. The B-address register is used on an unoverlapped command, and the E-address register is used on an overlapped command.

The Op-modifier character reads out at I11 time and is set into the Op-modifier register. For the card reader Op code, the Op-modifier character is an R to specify a read (or input) operation.

If the command is overlapped, the Op register, Op-modifier register, and the I/O channel-select register must be free to accept the characters of the next instruction word. The information in these registers is translated and stored in the E-channel move or load, and input or output latches in order to control the execution of the I/O command, and the registers are freed for the following instruction.



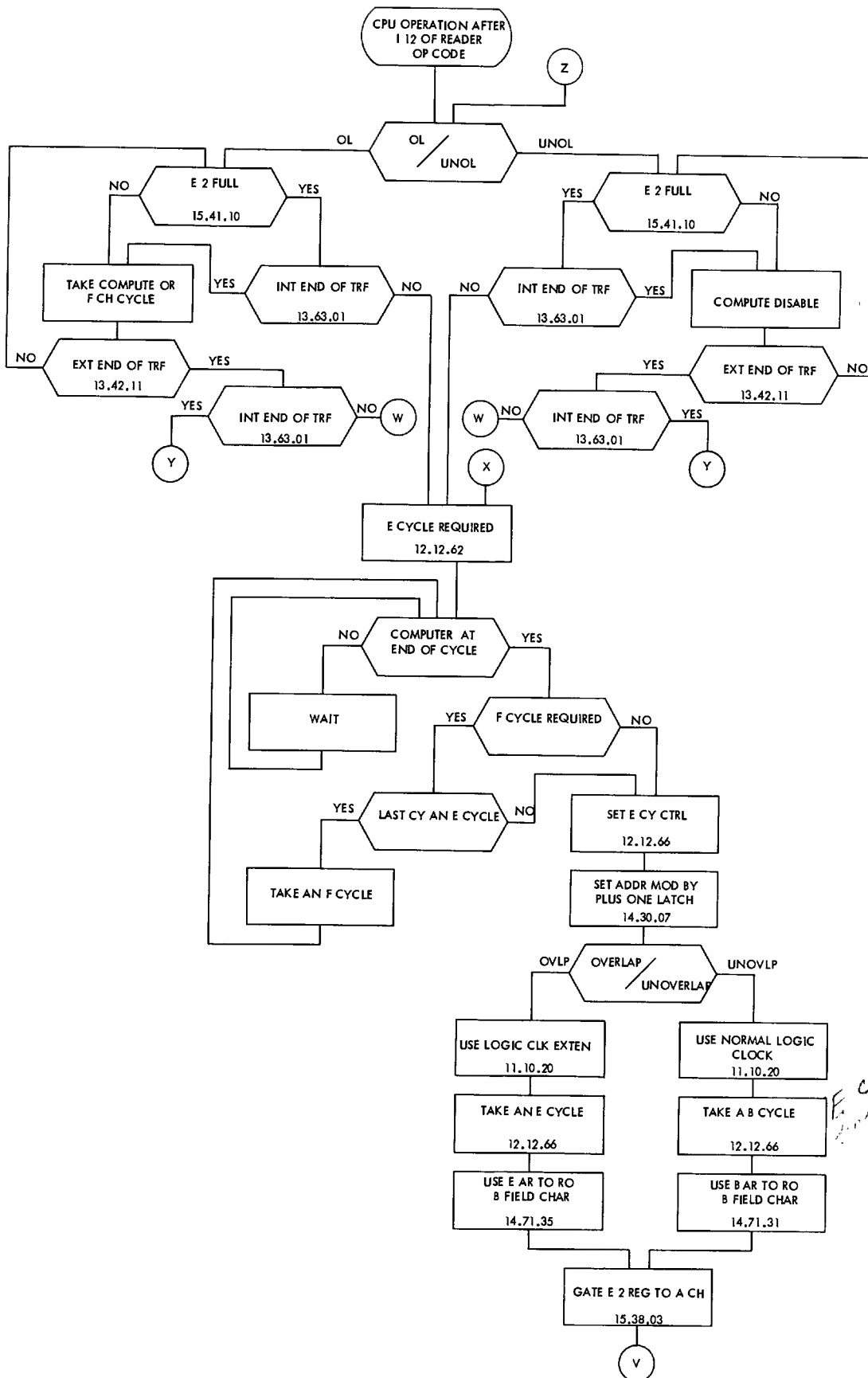


Figure 3.2-8B Card Read Op Code

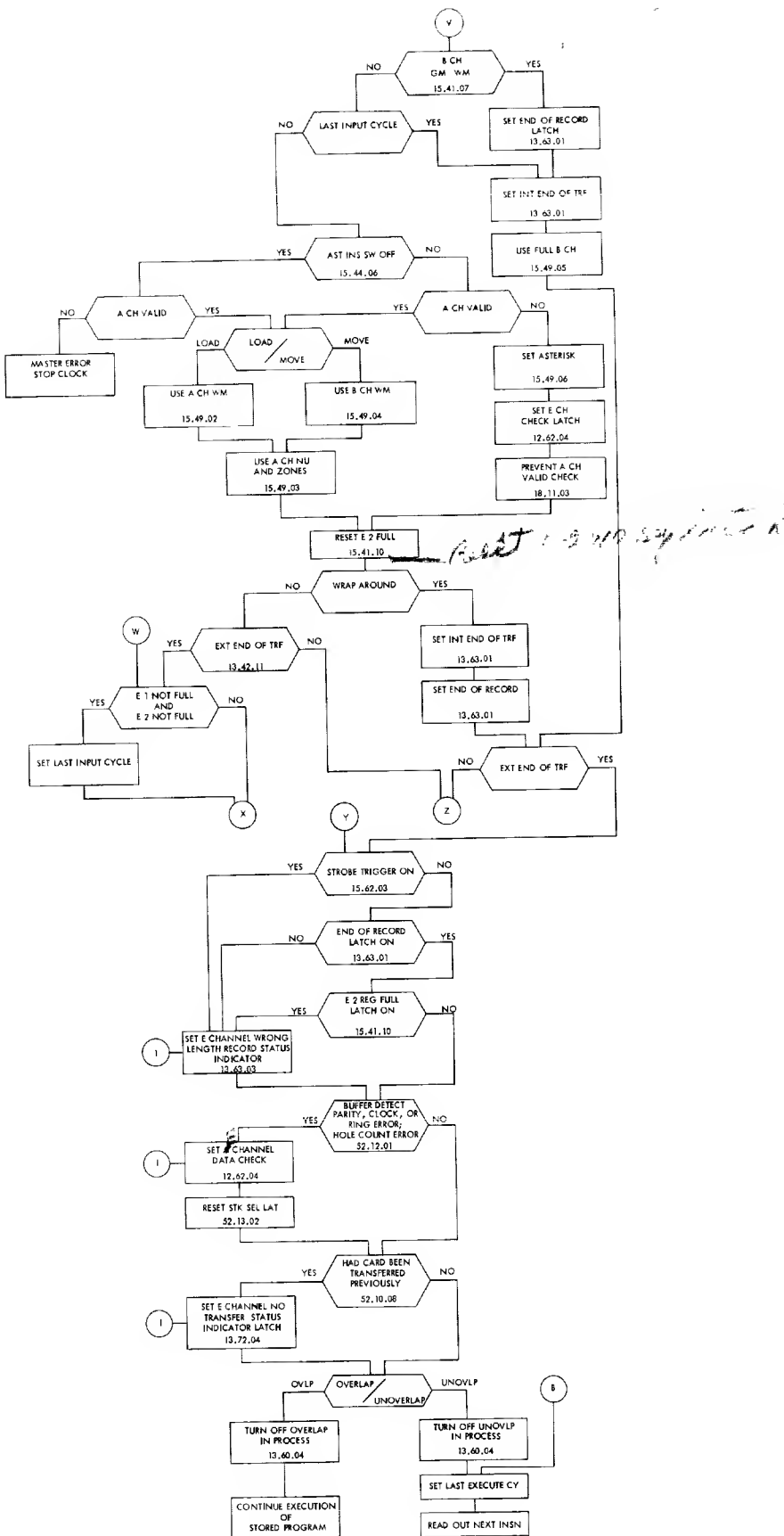


Figure 3.2-8C Card Read Op Code

This takes place at I12 or the last instruction read-out cycle. The status sample A-pulse develops at I12 for setting the status indicators to reflect the card-reader status. If the reader is not ready, the E-channel not-ready indicator sets. If the reader is busy, the E-channel busy indicator sets. If the last card is stacked in an end-of-file condition, the E-channel condition indicator sets. If any set at this time, the CPU does not execute the instruction, but instead it goes on to the next instruction in sequence and leaves the status indicators set. If no indicators are set by the status sample A-pulse, the read Op code can be executed.

Either the overlap or unoverlap in-process latch is set and reader-storage circuitry is conditioned to send its contents to the CPU by a request for a read-transfer scan. When the scan request is granted, the unit-number character is analyzed and reader circuits are set up accordingly. If the character is a 9, the reader clutch magnet is not energized and no stack select latches are set. However, if the character is not a 9, the read clutch magnet is energized, even though the read transfer has not yet taken place. This is possible because a certain amount of time is involved between the time when the clutch is energized and when the card begins to read into the buffer. This is sufficient time to complete the read transfer.

During each read-storage memory cycle, a character reads out and is sent to the CPU where it is set into the E1 data register. Transfer of the character from E1 to E2 is controlled by the word-separator control circuit. If the instruction is given in the load mode, a single word-separator character from the reader places a word mark over the following character. A pair of word-separator characters causes a single word-separator character to be written in storage. If the instruction is given in the move mode, a word-separator character is stored as a word separator character in the CPU.

If the command is unoverlapped, the CPU takes an E-cycle only when necessary to store the E2 character. During the time when E2 is not full, the CPU logic clock is disabled. If the command is overlapped, the CPU stores the E2 characters with the E-cycles, and continues to process subsequent instructions during the time when E2 is not full.

Once the read-transfer scan is initiated, the read-storage-scan circuitry runs continuously to read out each position of read storage and to send it to the CPU. The transfer continues until all 80 positions of read storage are transferred.

When a group-mark with a word-mark is detected in a storage position, or if a wrap-around condition is recognized, the internal transfer is ended and the CPU waits for the external end-of-transfer. If the external end-of-transfer is recognized before the internal end-of-transfer, an extra cycle is taken after E1 and E2 data registers are emptied in order to look for the group-mark with a word-mark. During this last input cycle, the internal end-of-transfer is forced whether or not the group-mark with a word-mark is detected.

With both internal and external transfers complete, a status sample B-pulse tests whether there are any transfer errors, and whether the record length is correct. Any reader-storage error during the transfer causes the E-channel data-check status-indicator to be set. A wrong-length record sets the E-channel wrong-length-record status indicator. In addition, the E-channel no-transfer status indicator is set if this card was previously transferred to the CPU. Once the indicators

are set, the CPU is released to continue with the stored program. The read clutch was previously energized; when the clutch engages, a card feed cycle is taken. An error during the transfer causes the stack-select latches to be reset so that the card in error selects into the N/R pocket.

## Circuits

Common Op-code grouping lines are:

Percent type op codes  
 Not addr dbl op codes  
 2 addr plus mod op codes  
 2 address op codes  
 Address type op codes  
 No index on 1st addr ops  
 No branch op codes  
 M or L op codes

1. Gate the unit-number character to the E2 data register.

<u>Signal</u>	<u>Control</u>	<u>Logic</u>
Sw B-ch to A-reg	1-cycle Logic gate D 2nd clock pulse 3	15. 38. 01
Use A-ch nu	I-cycle	15. 49. 03
Use A-ch zones	I-cycle (not) 1401 I-ring 5 or 10 time	15. 49. 03
Use A-ch WM	I-cycle	15. 49. 02
Set E1 reg	Logic gate A I-ring 5 time I-cycle 1 M or L op codes E-ch sel unit 1 Percent or com1 at	15. 62. 04
Set E2	E1 reg full (not) E2 reg full	15. 62. 04

2. Gate the E2 register character to the reader circuitry in the integrated synchronizer either to select a stacker and feed or to prevent a card from feeding.

<u>Signal</u>	<u>Control</u>	<u>Logic</u>
CPU to I/O sync___bit	E2 reg___bit	15. 60. 31-34
CPU to I/O bit	CPU to I/O sync___bit	51. 40. 10
Strobe latch (off)	Time 020-030 (not) Trans scan	51. 40. 43



<u>Signal</u>	<u>Control</u>	<u>Logic</u>
Set read feed latch	(not) 1401 mode Strobe latch (off) I/O to CPU trans Sel unit 1 (not CPU to I/O 8 and 1-bits)	51. 40. 43
Rd stk sel	Set read feed latch	51. 40. 50
Stack 1 or 2	Rd stk sel CPU to I/O bit___	52. 13. 02
Rd feed gate	Set read feed latch	52. 10. 08
Rd clutch	Rd feed gate Proc feed (CB timing)	52. 10. 09

3. Set the E-channel not-ready indicator, if the reader is not ready.

<u>Signal</u>	<u>Control</u>	<u>Logic</u>
Rd not ready to drive	Integrated buffer (off-line mode) CE rd trans gate CE read ready latch (off)	52. 11. 01
(not) Rd ready to CPU	Rd not ready to drive (not) Reader end-of-file	52. 11. 02
(not) Buffer ready	Rd ready to CPU Select unit 1	51. 40. 18
(not) E-ch ready bus	(not) Buffer ready E-ch sel any buffer	12. 62. 01
E-ch status sample A	M or L op codes I-ring 12 time (not) E-ch sel unit F I-cycle Percent or coml at (not) File op Logic gate D	13. 65. 06
E-ch not ready	E-ch status sample A (not) E-ch ready bus	12. 62. 01

4. Set the E-channel busy-status indicator, if the reader is busy.

<u>Signal</u>	<u>Control</u>	<u>Logic</u>
Reader busy	Rd feed Rd feed latch (on) 1401 read latch (on)	52. 10. 08
Buffer busy	Reader busy Select unit 1	51. 40. 19
E-ch busy bus	Buffer busy E-ch select any buffer	12. 62. 02

<u>Signal</u>	<u>Control</u>	<u>Logic</u>
E-ch busy	E-ch status sample A E-ch ready bus E-ch busy bus	12. 62. 02

5. Set the E-channel-condition indicator, if the reader end-of-file condition has been reached.

<u>Signal</u>	<u>Control</u>	<u>Logic</u>
Encoder	Rd reg Rd scan	52. 10. 05
E OF delay latch	Encode (end-of-file key pressed, hopper empty and number 1 card lever open)	52. 11. 01
E OF gate	Read feed latch or 1401 read latch	52. 10. 08
E-of-F	E OF delay latch E OF gate	52. 11. 01
Buffer conditions	E-or-F Select unit 1	51. 40. 20
E-ch condition bus	Buffer condition E-ch select any buffer	12. 62. 03
E-ch condition	E-ch condition (not) File op E-ch status sample A (not) E-ch select tape data E-ch ready bus (not) E-ch busy bus (not) 2 char only op codes (not) E-ch select unit to input	12. 62. 04

6. If any status indicator is ON, go to the next instruction.

<u>Signal</u>	<u>Control</u>	<u>Logic</u>
E-ch correct-length record	Set E-ch condition or Set E-ch busy or Set E-ch not ready	13. 63. 03
E-ch any status (on)	E-ch correct-length record	13. 72. 05
Last execute cycle I/O	E-ch any status (on) Last insn RO cycle (not) File op I/O percent latch	13. 65. 07
Last execute cycle	Last execute cycle I/O	12. 12. 51

7. If the condition latch is ON, reset the end-of-file latch.

<u>Signal</u>	<u>Control</u>	<u>Logic</u>
Reset select buffer latches	E-ch condition E-ch status sample A-delay E-ch select any buffer	13. 70. 04
Reset CPU	Reset select buffer latches	51. 40. 06
End-of-file latch (off)	Reset CPU Select unit 4	52. 11. 01

8. Request a read-transfer scan.

<u>Signal</u>	<u>Control</u>	<u>Logic</u>
Ready to buffer	(not) Clock stopped E-ch in-process E-ch select any buffer E-ch input mode	13. 70. 04
Ready	Ready to buffer	51. 40. 04
Rd trans req	Ready Rd ready (not) Reader busy Select unit 1	52. 10. 09
Rd priority gate	Rd trans req	52. 10. 07
Rd pr req trigger (on)	Rd priority gate Time 080-090	52. 10. 01
Gate on rd scan tr	Rd pr req tr (not) Some scan (not) Higher priority scan req	52. 10. 01
Rd scan	Gate on rd scan tr Time 030-040	52. 10. 02

9. Read out the addressed read-storage position.

<u>Signal</u>	<u>Control</u>	<u>Logic</u>
Time pl 1-latch(on)	Read scan (not) Single cycle mode	51. 30. 02
Time pl 2-latch(on)	Time pl 1-latch (on) Time 000-010	51. 30. 02
Timing pulse gate	Time pl 2-latch(on)	51. 30. 02
Rd pulse L-latch (on)	Time 000-010 Not clock 2	51. 30. 05
Rd pulse L-latch (off)	Time 040-050	51. 30. 05
Rd pulse S-latch (on)	Time 010-020	51. 30. 05

<u>Signal</u>	<u>Control</u>	<u>Logic</u>
Rd pulse 5-latch (off)	Time 040-050	51. 30. 05
Read pulse 1 010-040	Rd pulse 5-latch Timing pulse gate	51. 30. 05
Read pulse 2 005-040	Rd pulse L-latch Timing pulse gate	51. 30. 05

10. Set the strobe trigger.

<u>Signal</u>	<u>Control</u>	<u>Logic</u>
Trans scan	Rd trf scan	51. 40. 40
5strobe latch	Trans scan Time 020-030	51. 40. 43
I/O to CPU trans	Rd trf scan	51. 40. 40
Buffer strobe	I/O to CPU trans Time 080-090 5strobe latch	51. 40. 43
E-ch strobe trigger	Buffer strobe E-ch select any buffer	15. 62. 03

11. Set the E1 register; set the E1 full latch; reset the strobe trigger.

<u>Signal</u>	<u>Control</u>	<u>Logic</u>
Set E1	2nd clock pulse E-ch strobe trigger E-ch input mode (not) E1 reg full	15. 62. 04
E-ch strobe trigger (off)	E-ch input mode Set E1 reg	15. 62. 03
E1 full	Set E1	15. 41. 10

12. Analyze the E1 input character and set up controls for the E1 to E2 transfer.

<u>Signal</u>	<u>Control</u>	<u>Logic</u>
E-ch select 7-bit unit	E-ch select unit 1	13. 62. 03
E-ch word-separator mode	E-ch select 7-bit unit E-ch load mode	15. 62. 02
E-ch word separator	E1 input $C \cdot B \cdot 4 \cdot 2 \cdot 1 \cdot A \cdot B$ -bits	15. 41. 01
E1 reg word Separator latch (on)	E-ch word separator E-ch word-separator mode Set E1	15. 41. 11

13. Transfer E1 to E2. If the instruction is given the move mode, set E2 with E1 BCD.

<u>Signal</u>	<u>Control</u>	<u>Logic</u>
E-ch reset	I-ring 4 time Percent or com1 at Logic gate C	15. 41. 12
(not) E1 reg word separator	E-ch reset	15. 41. 11
Copy E1 BCD to E2	(not) E1 reg word separator	15. 62. 06
E2 reg full latch (off)	E-ch reset	15. 41. 10
Set E2 reg	E1 reg full latch (on) E2 reg full latch (off)	15. 62. 04
E2 reg full latch (on)	Set E2 reg (not) E1 reg word separator	15. 41. 10
E1 reg full latch (off)	E-ch input mode Set E2 reg	15. 41. 10

14. Transfer E1 to E2. If the instruction is given in the load mode, and E1 reg not-word-separator latch is ON, copy E1 BCD to E2.

<u>Signal</u>	<u>Control</u>	<u>Logic</u>
(not) E1 reg word separator	E-ch not-word-separator (not) E1 WM bit Set E1 reg	15. 41. 11
Copy E1 BCD to E2	(not) E1 reg word separator	15. 62. 06
E2 reg full latch (on)	Set E2 reg E1 reg not-word-separator	15. 41. 10
E1 reg full latch (off)	E-ch input mode Set E2 reg	15. 41. 10

15. Transfer E1 to E2. If the instruction is given in the load mode with a word-separator character in E1 and with no word separator in E2, generate a word-separator character and set it into E2 with a C-bit. E2 full is not set because the next character from read storage must be combined with the E2 character before the contents of E2 can be stored.

<u>Signal</u>	<u>Control</u>	<u>Logic</u>
Set E2 word separator	Set E2 reg (not) Gate console to assembly E1 reg word separator (not) E2 reg word separator	15. 62. 06
Copy E1 WM·C-bit	(not) E2 reg word separator E-ch input mode	15. 62. 06
Set E2 reg delayed	Set E2 reg (off)	15. 62. 04
E2 reg word separator	Set E2 reg delayed E-ch word-separator mode E1 reg word separator	15. 41. 11

16. Transfer E1 to E2. If the instruction is given in the load mode with a word-separator character in E2 and with no word separator in E1, copy E1 BCD and generate a word mark in E2.

<u>Signal</u>	<u>Control</u>	<u>Logic</u>
Copy inv E1 WM·C-bit	E2 reg word separator (not) E1 reg word separator E-ch input mode (not) Gate console to assembly	15. 62. 06
Copy E1 BCD to E2 reg	E2 reg word separator	15. 62. 06
E2 reg full latch (on)	Set E2 reg E1 reg not word separator	15. 41. 10

17. Transfer E1 to E2. If the instruction is given in the load mode with a word-separator character in E1 and E2, set E2 with E1 BCD and C-bit.

<u>Signal</u>	<u>Control</u>	<u>Logic</u>
Copy E1 BCD to E2 reg	E2 reg word separator	15. 62. 06
Copy E1 WM·C-bit	E2 reg word separator E1 reg word separator E1 reg not WM bit	15. 62. 06

18. Initiate a storage cycle in the CPU whenever E2 is full (E2 full latch ON). If the command is unoverlapped, take an E-cycle and read out the BAR.

<u>Signal</u>	<u>Control</u>	<u>Logic</u>
E-cycle required	E-ch in-process E-ch output mode (not) E-ch int end-of-transfer E1 reg full latch (off) (not) E-cycle · any last gate	12. 12. 62
(Start logic clock)	E-cycle required E-ch unovlap in-process	11. 10. 10
E-cycle ctrl	E-cycle required 2nd clock pulse Logic gate Z (not) E-ch int end-of-transfer (not) F-cycle required	12. 12. 66
E-cycle	E-cycle control Logic gate B or S	12. 12. 02
RO BAR	E-cycle ctrl E-ch unovlap in-process (not) Console inhibit AR RO (not) E-ch 2nd address · unovlp (not) 1st I/O cycle control	14. 71. 31

19. Initiate a storage cycle in the CPU whenever E2 is full (E2 full latch ON). If the command is overlapped, take an E-cycle and read out the EAR.

<u>Signal</u>	<u>Control</u>	<u>Logic</u>
E-cycle required	E-ch in-process E-ch output mode (not) E1 reg full (not) E-ch int end-of-transfer (not) E-cycle · any last gate	12. 12. 62
(Start logic extension clock)	E-cycle required E-ch overlap in-process	11. 10. 20
E-cycle ctrl	E-cycle required 2nd clock pulse Logic gate Z (not) E-ch int end-of-transfer (not) F-cycle required	12. 12. 66
E-cycle	E-cycle ctrl Logic gate B or S	12. 12. 66
RO EAR	E-cycle ctrl E-ch ovlp in-process (not) Console inhibit AR RO (not) E-ch 2nd Addr ovlp	14. 71. 35

20. Gate the E2 character to the A-channel

<u>Signal</u>	<u>Control</u>	<u>Logic</u>
Gate E2 data reg to A-ch	E-ch input mode E-cycle (not) Control reg disable	15. 38. 03

21. End the internal transfer, if the B-channel character is a group-mark with a word-mark.

<u>Signal</u>	<u>Control</u>	<u>Logic</u>
B-ch group mark · WM	B-ch 1 · 2 · 4 · 8 · A · B · WM	15. 41. 07
E-ch end-of-record latch	E-cycle B-ch group mark · WM Logic gate F R symbol op modifier (not) 1401 card print in-proc	13. 63. 01

22. End the internal transfer, if a wrap-around condition occurs.

<u>Signal</u>	<u>Control</u>	<u>Logic</u>
E-ch end-of-record latch	(not) E-ch last input cycle Wrap-around conditions Logic gate F or W 2nd clock pulse 2 E-cycle	13. 63. 01

<u>Signal</u>	<u>Control</u>	<u>Logic</u>
E-ch int end-of-transfer	E-cycle (not) E-ch last input cycle Wrap-around conditions Logic gate F or W 2nd clock pulse 2	13. 63. 01

23. End the internal transfer, if the external end-of-transfer occurs before the internal end-of-transfer. Take on extra input cycle to look for the group-mark with a word-mark, then end the internal transfer.

<u>Signal</u>	<u>Control</u>	<u>Logic</u>
E-ch last input cycle	E-ch ext end-of-transfer (not) E1 reg full (not) E2 reg full E-cycle E-ch input mode Logic gate C or T	13. 63. 02
E-ch int end-of-transfer	E-ch last input cycle Logic gate F or W	13. 63. 01

24. Develop the assembly controls. If the asterisk insert switch is ON and the A-channel character is invalid, set an asterisk.

<u>Signal</u>	<u>Control</u>	<u>Logic</u>
In cy GM·WM ctrl	B-ch not group mark · WM Input cycle (not) any last input cycle	15. 49. 02
A-ch invalid	(Invalid bit configuration)	18. 11. 03
Set asterisk	In cy GM·WM ctrl A-ch invalid (not) Asterisk ins console sw (off)	15. 49. 06

25. Develop the assembly controls. If the asterisk insert switch is OFF and the A-channel character is invalid, stop the clock.

<u>Signal</u>	<u>Control</u>	<u>Logic</u>
Master error	A-channel VC error Logic gate A or R	18. 14. 08
Stop latch	Master error	12. 15. 04

26. Develop the assembly controls. If the instruction is given in the load mode and there is a valid A-channel character, use the A-channel numerical bits, word mark bit, and zone bits.

<u>Signal</u>	<u>Control</u>	<u>Logic</u>
A-ch valid + ast switch (off)	A-ch valid or Asterisk ins console sw (off)	15. 49. 02



<u>Signal</u>	<u>Control</u>	<u>Logic</u>
E-ch select odd parity unit	E-ch select unit 1	13. 60. 03
Odd-parity cycle	E-ch select odd-parity unit E-cycle	13. 60. 02
Use A-ch zones	In cy GM·WM ctrl A-ch valid + ast switch (off) Odd-parity cycle	15. 49. 03
Use A-ch nu	In cy GM·WM ctrl A-ch valid + ast switch (off) Odd-parity cycle	15. 49. 03
Use A-ch WM	In cy GM·WM ctrl A-ch valid + ast switch (off) Load cycle	15. 49. 02

27. Develop the assembly controls. If the instruction is given in the move mode and there is a valid A-channel character, use the A-channel numerical and zone bits and the B-channel word-mark bit.

<u>Signal</u>	<u>Control</u>	<u>Logic</u>
Use A-ch zones	In cy GM·WM ctrl A-ch valid + ast switch (off) Odd-parity cycle	15. 49. 03
Use A-ch nu	In cy GM·WM ctrl A-ch valid + ast switch (off) Odd-parity cycle	15. 49. 02
Use B-ch WM	Move cycle Input cycle	15. 49. 04

28. Develop the assembly controls. If the internal end-of-transfer is detected, use the full B-channel character.

<u>Signal</u>	<u>Control</u>	<u>Logic</u>
I/O grp mk·WM stop ctrl	E-ch ovlp in-process E-cycle or E-ch unovlp in-process E-cycle R-symbol op modifier	13. 65. 04
Use full B-ch	I/O grp mk·WM stop ctrl B-ch group mark·WM Input cycle (not) 1401 card print in-proc	15. 49. 05

29. Develop the assembly controls. If the CPU is in the last input cycle, use the full B-channel character.

<u>Signal</u>	<u>Control</u>	<u>Logic</u>
Any last input cycle	E-ch last input cycle	13. 63. 02

<u>Signal</u>	<u>Control</u>	<u>Logic</u>
Any last in cycle not 1401	Any last input cycle (not) 1401 mode	15. 49. 05
Use B-ch nu	Any last in cycle not 1401	15. 49. 05
Use B-ch zones	Any last in cycle not 1401	15. 49. 05
Use B-ch WM	Any last in cycle not 1401	15. 49. 04

30. Reset the E2 full latch after each input cycle.

<u>Signal</u>	<u>Control</u>	<u>Logic</u>
Reset E2 full latch	Any last gate E-ch cycle In cy GM-WM ctrl 2nd clock pulse (inverted)	15. 62. 05

31. With both internal and external transfers ended, develop a sample pulse for setting the E-channel status indicators.

<u>Signal</u>	<u>Control</u>	<u>Logic</u>
E-ch int end-of-tfr delayed	Logic gate Z E-ch int end-of-transfer	13. 65. 05
E-ch status sample B	E-ch ext end-of-transfer E-ch int end-of-trf delayed Logic gate Z 2nd clock pulse 1 (inverted)	13. 65. 05

32. Set the E-channel wrong-length-record status indicator, if the record length is incorrect.

<u>Signal</u>	<u>Control</u>	<u>Logic</u>
E-ch wrong-length record	E-ch status sample B E-ch strobe trigger or E2 reg full or (not) E-ch end-of-record latch	13. 63. 03

33. Set the E-channel data-check status indicator, if any error occurred during the transfer.

<u>Signal</u>	<u>Control</u>	<u>Logic</u>
Read error	Invalid card code or Pwrd clock error or Ring error to error latch or Parity 1 or	52. 12. 01

<u>Signal</u>	<u>Control</u>	<u>Logic</u>
	Parity 2 or Hole-count error	
Buffer error	Read error Select unit 1	51. 40. 21
E-ch check bus	Buffer error E-ch select any buffer	12. 62. 03
E-ch check	E-ch check bus E-ch status sample B	12. 62. 04

34. Reset the reader-stacker select latches, if an error occurred during the transfer.

<u>Signal</u>	<u>Control</u>	<u>Logic</u>
(Read stack 1 and 2 latches reset)	Read error	52. 13. 02

35. Set the E-channel no-transfer status-indication, if this record was transferred previously.

<u>Signal</u>	<u>Control</u>	<u>Logic</u>
(not) Multi read feed	Rd trans req (not) Rd feed	52. 10. 08
(not) Gated multi read feed	(not) Multi read feed (not) Ready	52. 10. 08
Buffer no-trans cond	(not) Gated multi read feed Reset CPU or GO Buffer end-of-transfer Sel unit 1 (not) Stk sel	51. 40. 12
E-ch no-transfer latch	Buffer no-Xfer cond E-ch status sample B E-ch select unit 1 (not) E-ch busy bus E-ch ready bus	13. 72. 04

36. Continue with the stored program. If the command is unoverlapped, set last execute cycle and reset the unoverlap-in-process latch.

<u>Signal</u>	<u>Control</u>	<u>Logic</u>
Last-execute cycle I/O	E-ch unovlp-in-process (not) 1401 card or print op code E-ch status sample B	13. 65. 07
E-ch second sample B	E-ch status sample 3 2nd clock pulse (inverted)	13. 65. 05

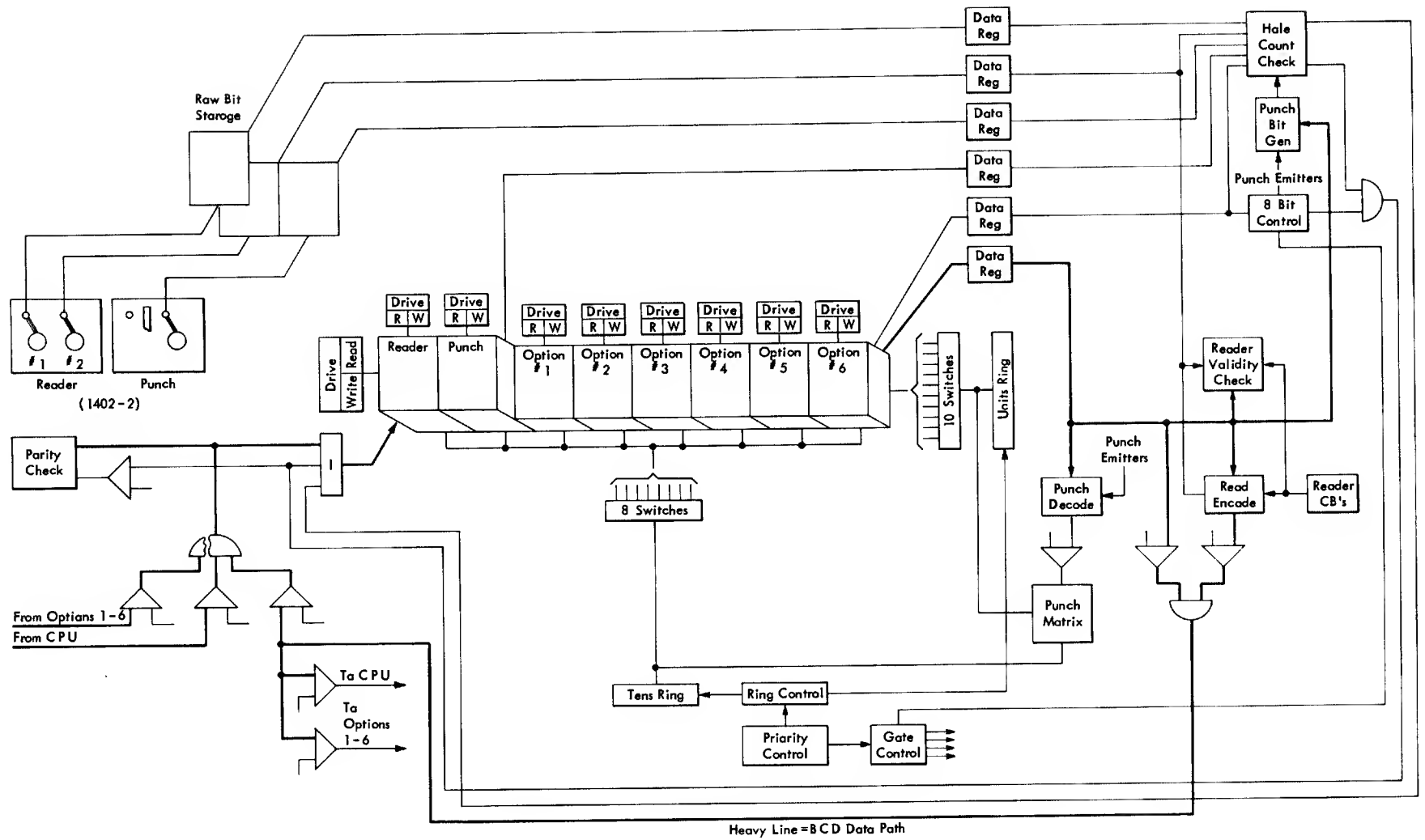


Figure 3.3-1 IBM 1414 Integrated Synchronizer Data Flow

<u>Signal</u>	<u>Control</u>	<u>Logic</u>
E-ch status sample B-delay	E-ch second sample B 2nd clock pulse (inverted)	13. 65. 05
(Reset) E-ch unovlp-in- process	E-ch status sample B-delay	13. 60. 04

37. Continue with the stored program. If the command is overlapped, reset the overlap-in-process latch.

<u>Signal</u>	<u>Control</u>	<u>Logic</u>
(Reset) E-ch ovlp-in- process	E-ch status sample B-delay	13. 60. 04

### 3. 3. 00 PAPER-TAPE-READER ATTACHMENT

The IBM 1011 Paper Tape Reader is used in conjunction with the integrated synchronizer to provide a buffered-paper-tape input to the IBM 1410 Data Processing System. There are thirteen lines that connect the 1011 and the integrated synchronizer. The names and functions of these lines are:

<u>Line</u>	<u>Function</u>
Paper-tape-ready	Output line from the 1011. Indicates that power is ON, that tape is properly loaded, and that the start key was pressed.
PT parity error	Output line from the 1011. Indicates an 8-channel tape-parity error.
End-of-record	Output line from the 1011. Indicates that the end-of-record character was detected.
Data output (7 lines)	Output lines from the 1011 output data register (register B).
Data sync	Output line from the 1011. Indicates that a character is present on the data-output lines.
Character received	Input line to the 1011. Indicates that the character on the data-output lines was received and stored.
Go	Input line to the 1011 that requests a character.

### 3. 3. 01 Core Storage (Figure 3. 3-1)

The integrated synchronizer can have eight 80-position core-storage units. One of these storage units can be assigned to the paper-tape reader. Eighty characters from the paper-tape (PT) reader are stored in the PT storage unit. When the storage unit is filled, the contents can be transferred to the CPU. PT storage has eight

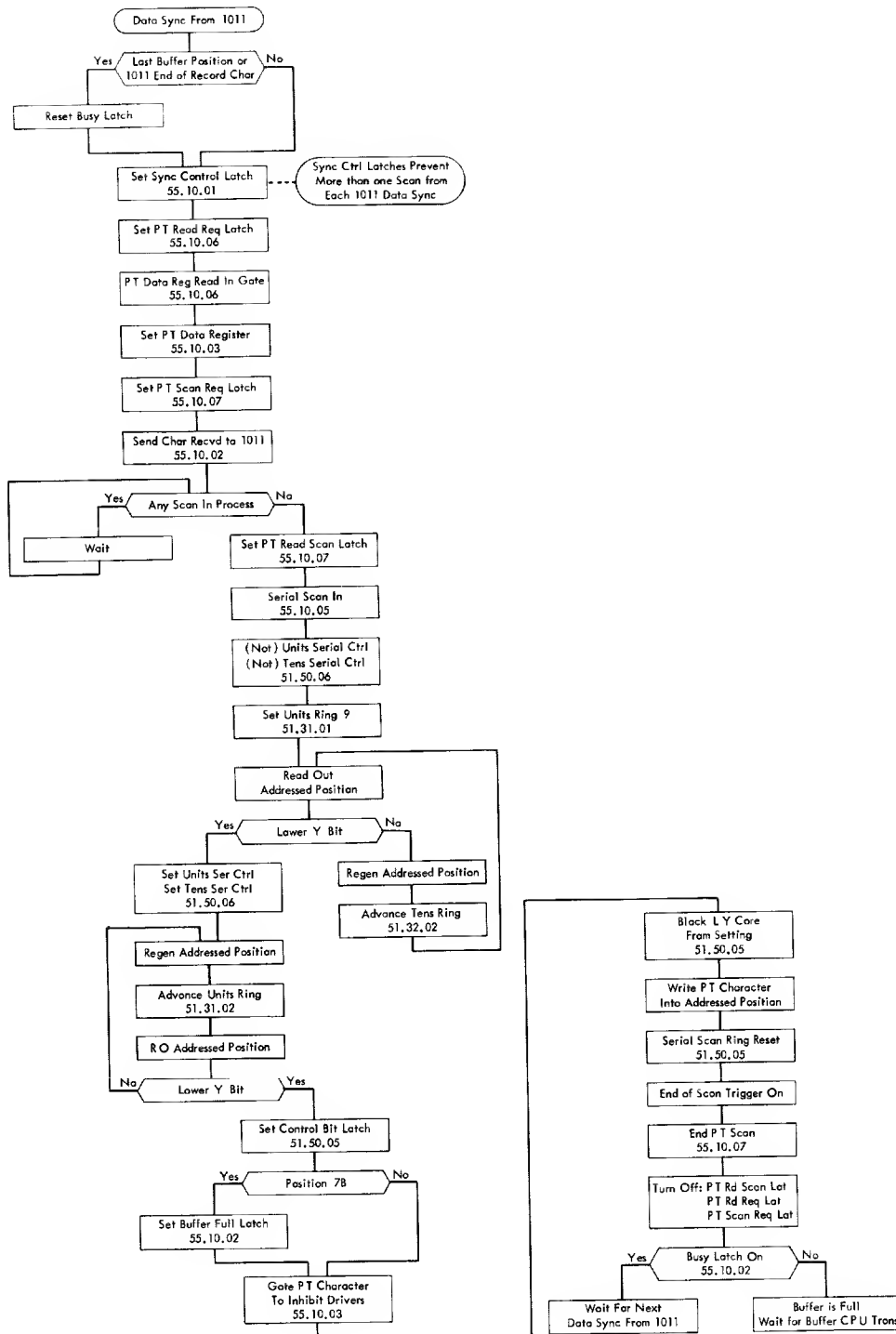


Figure 3.3-2 Paper-Tape Read-In

planes: seven planes store the characters in BCD form; the eighth, or control plane, locates the storage positions for incoming characters. When used with the integrated synchronizer, the PT reader uses the circuitry assigned to option 4.

PT storage uses the same addressing system, data registers, and clock as the other integrated synchronizer storage units.

### 3.3.02 Input-Serial-Scan Control

Once the go signal is received, the PT reader sends a character to PT storage every two milliseconds. Each time a character is received, the addressing rings locate the next storage position in sequence. The character is stored and the rings reset. As soon as the rings reset, they can be used to scan other storage units until the next PT character is received.

PT read-in begins with all eight-bit cores set ON. When a character is written into a PT storage position, the corresponding eighth-bit is set OFF. For example, assume that a PT read-in is in-process, and positions 1-23 have characters with no control bits in them. Positions 24-79 contain control bits. A data sync pulse from the PT reader requests a serial scan. The scan starts with the rings addressing position 09. Position 09 reads out, and because there is no control bit, the character is regenerated. The tens ring then advances and position 19 reads out. Again there is no control bit, so that the character is regenerated, and the tens ring advances to address position 29. When position 29 reads out, a control bit is detected. This indicates that the desired position is somewhere in the positions 20 to 29. The tens ring is now prevented from advancing, and the units ring is allowed to advance to address and to read out positions 20, 21, 22, and 23 before the next control bit is sensed. When position 24 reads out, presence of the control bit brings up the gate that allows the incoming character to be written into this position, during the write portion of the memory cycle. The control bit is not regenerated in order to indicate that this position is filled. The rings are then reset and the PT storage circuitry waits for the next data-sync pulse.

### 3.3.03 Paper-Tape Read-In (Figure 3.3-2)

Paper-tape read-in is initiated by the go signal from the integrated synchronizer. The go signal is generated with a power-on reset, or after a read transfer scan takes place. When the IBM 1011 receives the go signal, the drive mechanism is set in motion, and as each character becomes available at the output bit lines, a data-sync pulse is sent to the paper-tape storage circuitry. The data-sync pulse requests a serial scan to store the incoming character. Requesting the scan gates the character into a data register, and then signals the 1011 that the character is received.

After priority conditions are satisfied, a scan is initiated to locate the first blank position. When the correct position is located, the character is stored, and the rings are reset. After 80 data sync pulses are received, or if the 1011 signals an end of record, the go line is dropped to stop paper-tape reading. These conditions also reset the paper-tape-busy latch to allow a paper-tape-read-transfer instruction to be executed. Figure 3.3-3 shows the relative timings involved in storing a character.

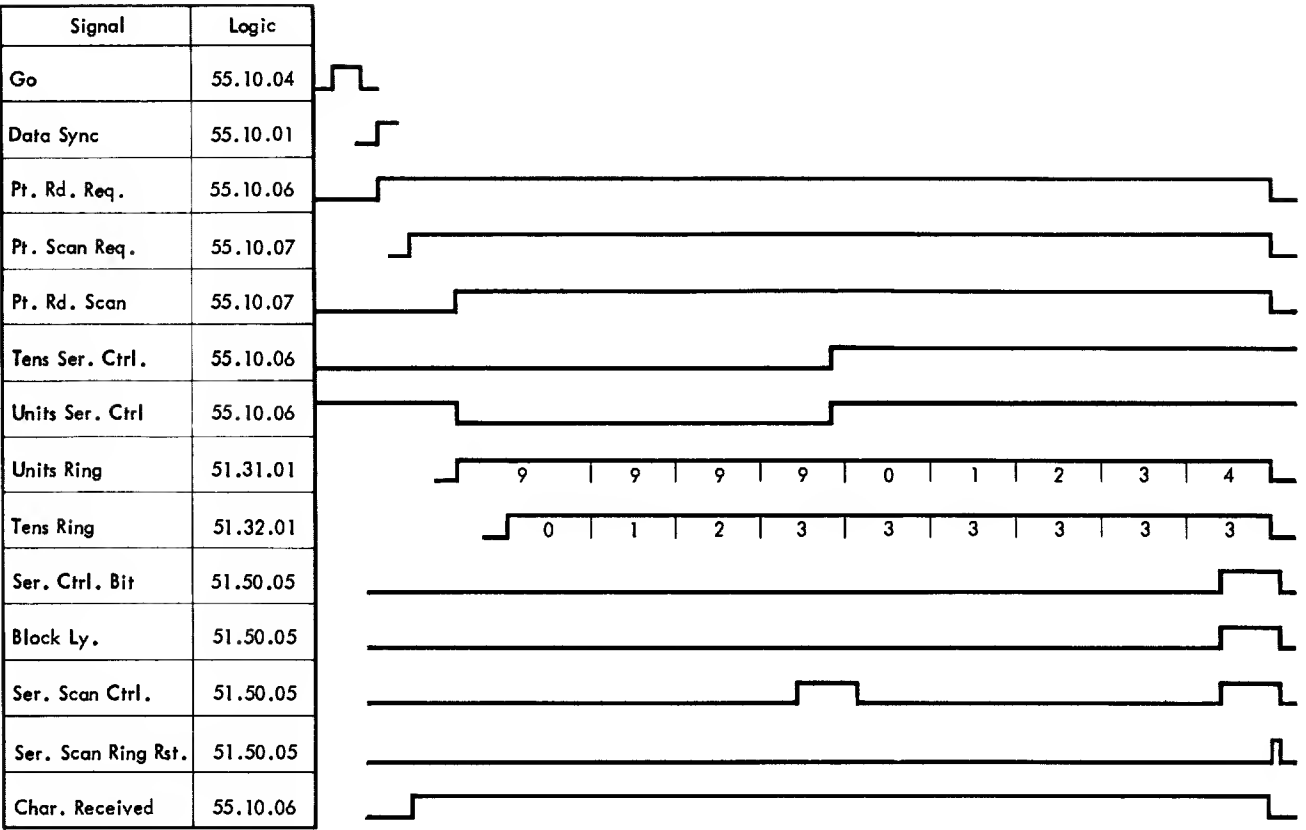


Figure 3.3-3 Paper-Tape Character Storage



### 3. 3. 04 Paper-Tape Reader Move or Load

#### Op Code Function (O XXX BBBB d)

The PT reader move or load Op code transfers the contents of PT storage to the CPU. The move Op code transfers the data exactly as it appears in PT storage. If the instruction is given in the load mode, a single word-separator character from PT storage causes a word-mark to be placed over the next character that is read into the CPU. Two successive word-separator characters from PT storage cause a single word-separator character to be placed in core storage in the CPU.

#### Operation (Figure 3. 3-4)

Instruction read-out causes the instruction word characters to be placed in various registers to control the execution of the card read Op code.

At I-Op time, the Op-code character reads out and is placed into the Op register. If the character is an M, the move mode is specified, and word-separator characters in PT storage move to CPU storage as word-separator characters. An L-Op code specifies the load mode to allow word marks to be generated in CPU storage from the reader-storage word-separator characters.

At I1 time, the I/O channel-select character reads out. The I-ring advances to I3, and the channel-select character is placed into the I/O channel-select register. The I/O interlock check is made at I3 to insure that the E-channel-status indicators were tested by the stored program since the last I/O command that used the E-channel. The check consists of testing the move or load latch. Because the RI $\ddagger$  resets these latches, they should both be OFF at I3. If either latch is ON at I3, an I/O interlock results. This produces a master error that stops the CPU.

At I4, the unit-select character reads out and is placed in the E-channel-unit select register, because the E-channel was selected at I3 time. Unit-select character P gates the PT reader status to the CPU. An E-channel reset develops at I4 to reset all control and status-indicator latches associated with this channel.

Unit number character O reads out and is placed into the E-channel-unit-number register at I5 time.

At I6 through I10, the B-address reads out and is placed in the B- and D-address registers. In addition, the B-address is placed into the E-address register on an overlapped command. The B-address register is used on an unoverlapped command, and the E-address register is used on an overlapped command.

The Op-modifier character reads out at I11 time and is set into the Op-modifier register. For the PT reader Op code, the Op-modifier character is an R to specify a read (or input) operation.

If this command is overlapped, the Op register, the Op-modifier register, and the I/O channel-select register must be free to accept the characters of the next instruction word. The information in these registers is translated and stored in the E-channel move or load, and input or output latches to control the execution of this I/O command and the registers are freed for the following instruction.



Figure 3.3-4A Paper-Tape Move or Load

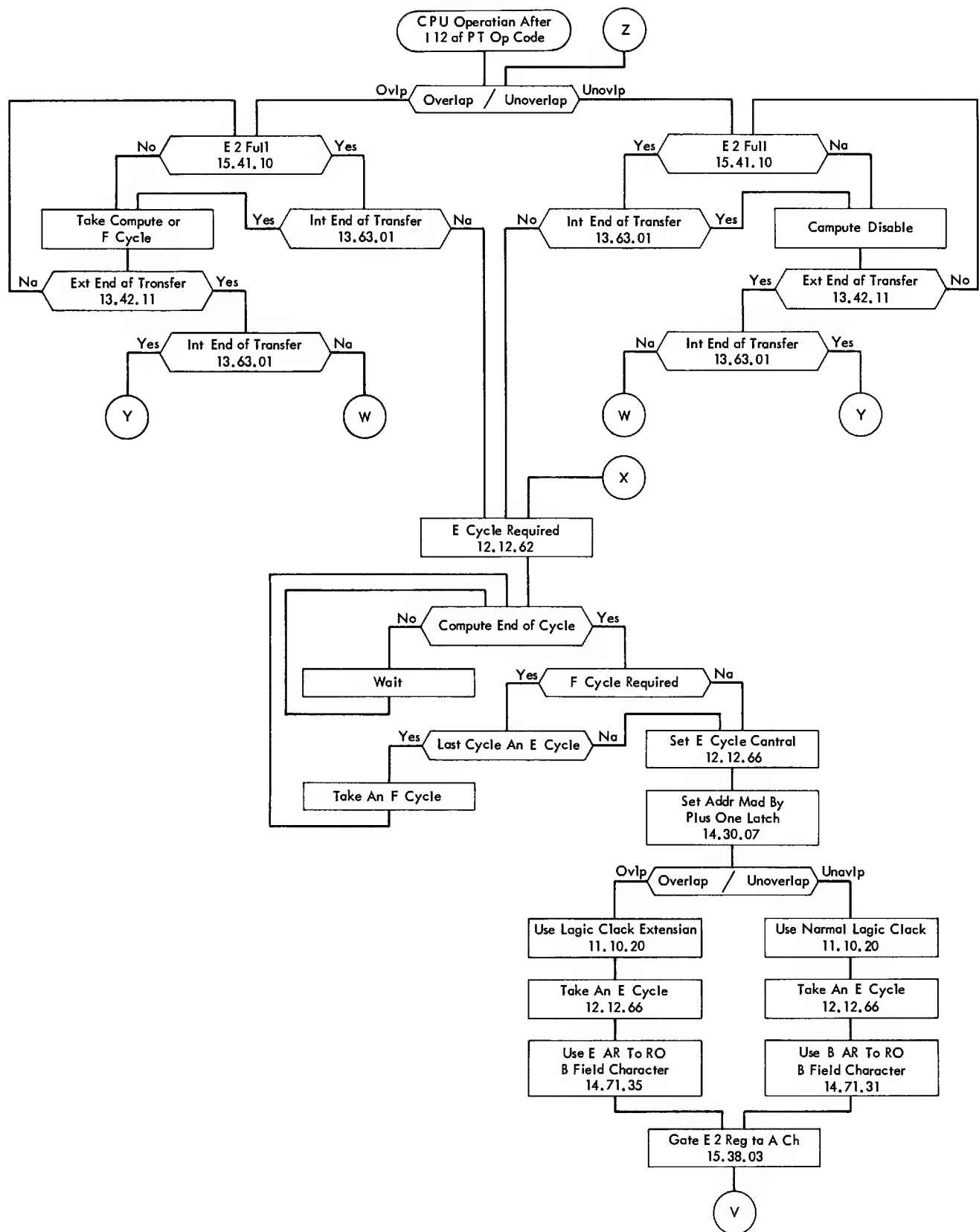


Figure 3.3-4B Paper-Tape Move or Load

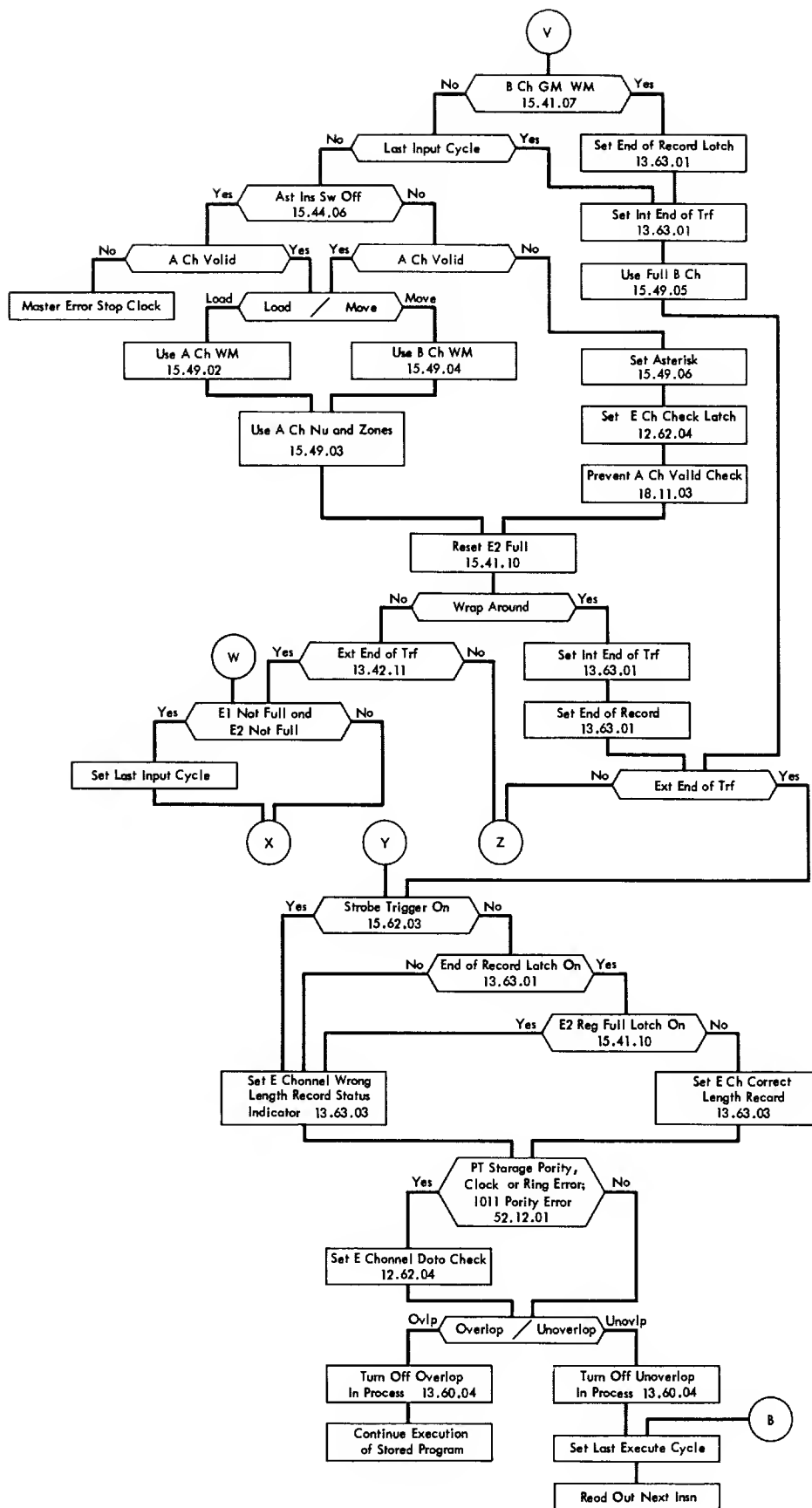


Figure 3.3-4C Paper-Tape Move or Load

This takes place at I12 or the last-instruction read-out cycle. The status sample A-pulse develops at I12 for setting the status indicators to reflect the PT reader status. If the PT reader is not ready, the E-channel-not-ready indicator is set. If the PT reader is busy, the E-channel-busy indicator is set. If either indicator is set at this time, the CPU does not execute this instruction. Instead it goes on to the next instruction, in sequence, and leaves the status indicators set. If no indicators are set by the status sample A-pulse, the PT move or load Op code can be executed.

Either the overlap or unoverlap-in-process latch is set and the PT storage circuitry is conditioned to send its contents to the CPU by a request for a PT read-transfer scan. When the scan request is granted, the PT storage reads out, and sends PT storage characters to the CPU.

During each PT storage-memory cycle, a character reads out and is sent to the CPU, where it is set into the E1 data register. Transfer of the character from E1 to E2 is controlled by the word-separator control circuit. If the instruction is given in the load mode, a single word-separator character from the reader places a word mark over the following character. A pair of word-separator characters causes a single word-separator character to be written in storage. If the instruction is given in the move mode, a word-separator character from PT storage is stored as a word-separator character in the CPU.

If the command is unoverlapped, the CPU takes an E-cycle only when necessary to store the E2 character. During the time when E2 is not full, CPU logic 1 is disabled. If the command is overlapped, the CPU stores the E2 characters with the E-cycles, and continues to process subsequent instructions during the time when E2 is not full.

Once the read-transfer scan is initiated, the PT storage-scan circuitry runs continuously to read out each position of PT storage and to send it to the CPU. The transfer continues until all 80 positions of PT storage are transferred. As each PT storage character reads out, a control bit (LY core plane) is placed in that position in preparation for the next PT read-in.

When a group-mark with a word-mark is detected in a storage position, or if a wrap-around condition is detected, the internal transfer is ended and the CPU waits for the external end-of-transfer. If the external end-of-transfer is detected before the internal end-of-transfer, an extra cycle is taken after the E1 and E2 data registers are emptied to search for the group-mark, with a word-mark. During this last input cycle, the internal end-of-transfer is forced whether or not the group-mark with a word-mark is detected.

With both internal and external transfers complete, a status sample B-pulse tests for transfer errors, and whether the record length was correct. Any PT storage error during the transfer causes the E-channel data-check status-indicator to be set. A wrong-length record sets the E-channel wrong-length record status indicator. In addition, a parity error in the IBM 1011 Paper Tape Reader (8-channel tape only) causes the E-channel data-check indicator to be set. Once the indicators are set, the CPU is released to continue with the stored program.

As soon as the transfer is complete, the PT storage circuitry sends the go signal to the PT reader in order to cause the next record to be loaded into PT storage.

## Circuits

Common Op-code grouping lines are:

Percent type op codes  
 Not addr dbl op codes  
 2 Addr plus mod op codes  
 2 Address op codes  
 Address type op codes  
 No-index on 1st addr ops  
 No-branch op codes  
 M or L op codes

1. Set the E-channel not-ready indicator, if the PT reader is not ready.

<u>Signal</u>	<u>Control</u>	<u>Logic</u>
(not) PT ready	(not) PT ready to adapter	55. 10. 01
PT rd not ready	(not) PT ready (or off-line CE mode)	55. 10. 02
PT rd op	Select unit P Read from buffer	55. 10. 04
(not) 14C5-A rdy status	PT rd not ready PT rd op	55. 10. 04
(not) Buffer ready	(not) 14C5 Rdr status	51. 40. 18
(not) E-ch ready bus	(not) Buffer ready E-ch sel any buffer	12. 62. 01
E-ch status sample A	M or L op codes I-ring 12 time (not) E-ch sel unit F I-cycle Percent or coml at (not) File op Logic gate D	13. 65. 06
E-ch not ready	E-ch status sample A (not) E-ch ready bus	12. 62. 01

2. Set the E-channel-busy indicator, if the PT reader is busy.

<u>Signal</u>	<u>Control</u>	<u>Logic</u>
Busy	End PT scan or PT busy or PT rd req	55. 10. 04
14C5A busy status	Busy PT rd op	55. 10. 04

<u>Signal</u>	<u>Control</u>	<u>Logic</u>
Buffer busy	14C5-A Busy status	51. 40. 19
E-ch busy bus	Buffer busy E-ch select any buffer	12. 62. 02
E-ch busy	E-ch status sample A E-ch ready bus E-ch busy bus	12. 62. 02

3. If any status indicator is ON, go to the next instruction.

<u>Signal</u>	<u>Control</u>	<u>Logic</u>
E-ch correct-length record	Set E-ch busy or Set E-ch not ready	13. 63. 03
E-ch any status (on)	E-ch correct-length record	13. 72. 05
Last execute cycle I/O	E-ch any status (on) Last insn RO cycle (not) File op I/O percent latch	13. 65. 07
Last execute cycle	Last execute cycle I/O	12. 12. 51

4. Initiate a PT read-transfer scan.

<u>Signal</u>	<u>Control</u>	<u>Logic</u>
Ready to buffer	(not) Clock stopped E-ch in-process E-ch select any buffer E-ch input mode	13. 70. 04
Ready	Ready to buffer	51. 40. 04
PT rd trans	Ready PT rd op	52. 10. 09
Gate on PT scan req	PT rd trans	55. 10. 06
PT scan req	Gate on PT scan req Time 080-090	55. 10. 07
PT rd scan	PT scan req (not) Same scan Time 030-040	55. 10. 07

5. Read out the addressed PT storage position.

<u>Signal</u>	<u>Control</u>	<u>Logic</u>
Time Pl 1 latch (on)	Opt no 4 scan (not) Single cycle mode	51. 30. 02
Time Pl 2 latch (on)	Time Pl 1 latch (on) Time 000-010	51. 30. 02

<u>Signal</u>	<u>Control</u>	<u>Logic</u>
Timing pulse gate	Time Pl 2 latch (on)	51. 30. 02
Rd pulse L latch (on)	Time 000-010 Not clock 2	51. 30. 05
Rd pulse L latch (off)	Time 040-050	51. 30. 05
Rd pulse S latch (on)	Time 010-020	51. 30. 05
Rd pulse S latch	Time 040-050	51. 30. 05
Read pulse 1 010-040	Rd pulse S latch Timing pulse gate	51. 30. 05
Read pulse 2 005-040	Rd pulse L latch Timing pulse gate	51. 30. 05

6. Set the strobe trigger.

<u>Signal</u>	<u>Control</u>	<u>Logic</u>
Trans scan	14C5-A trans scan	51. 40. 40
Strobe latch	Trans scan Time 020-030	51. 40. 43
I/O to CPU trans	Trans scan Read from buffer	51. 40. 40
Buffer strobe	I/O to CPU trans Time 080-090 Strobe latch	51. 40. 43
E-ch strobe trigger	Buffer strobe E-ch select any buffer	15. 62. 03

7. Set E1 register; set E1 full latch; reset the strobe trigger.

<u>Signal</u>	<u>Control</u>	<u>Logic</u>
Set E1	2nd clock pulse E-ch strobe trigger E-ch input mode (not) E1 reg full	15. 62. 04
E-ch strobe trigger (off)	E-ch input mode Set E1 reg	15. 62. 03
E1 full	Set E1	15. 41. 10

8. Analyze the E1 input character, and set up controls for the E1 to E2 transfer.

<u>Signal</u>	<u>Control</u>	<u>Logic</u>
E-ch select 7-bit unit	E-ch select unit 1	13. 62. 03
E-ch word-separator mode	E-ch select 7-bit unit E-ch load mode	15. 62. 02



<u>Signal</u>	<u>Control</u>	<u>Logic</u>
E-ch word separator	E1 input C · B · 4 · 2 · 1 · A · B-bits	1S. 41. 01
E1 reg word separator latch (on)	E-ch word separator E-ch word-separator mode Set E1	1S. 41. 11

9. Transfer E1 to E2. If the instruction is given in the move mode, set E2 with E1 BCD.

<u>Signal</u>	<u>Control</u>	<u>Logic</u>
E-ch reset	I-ring 4 time Percent or com1 at Logic gate C	1S. 41. 12
(not) E1 reg word separator	E-ch reset	1S. 41. 11
Copy E1 BCD to E2	(not) E1 reg word separator	1S. 62. 06
E2 reg full latch (off)	E-ch reset	1S. 41. 10
Set E2 reg	E1 reg full latch (on) E2 reg full latch (off)	1S. 62. 04
E2 reg full latch (on)	Set E2 reg (not) E1 reg word separator	1S. 41. 10
E1 reg full latch (off)	E-ch input mode Set E2 reg	1S. 41. 10

10. Transfer E1 to E2. If the instruction is given in the load mode, and the E1 reg not-word-separator latch is ON, copy E1 BCD to E2.

<u>Signal</u>	<u>Control</u>	<u>Logic</u>
(not) E1 reg word separator	E-ch not-word separator Set E1 reg	1S. 41. 11
Copy E1 BCD to E2	(not) E1 reg word separator	1S. 62. 06
E2 reg full latch (on)	Set E2 reg E1 reg not-word separator	1S. 41. 10
E1 reg full latch (off)	E-ch input mode Set E2 reg	1S. 41. 10

11. Transfer E1 to E2. If the instruction is given in the load mode with a word-separator character in E1 and no word separator in E2, generate a word-separator character and set it into E2 with a C-bit. The E2 full latch is not set because the next character from read storage must be combined with the E2 character before the contents of E2 can be stored.

<u>Signal</u>	<u>Control</u>	<u>Logic</u>
Set E2 word separator	Set E2 reg (not) Gate console to assembly E1 reg word separator (not) E2 reg word separator	1S. 62. 06

<u>Signal</u>	<u>Control</u>	<u>Logic</u>
Copy E1 WM · C-bit	(not) E2 reg word separator E-ch input mode	15. 62. 06
Set E2 reg delayed	Set E2 reg (off)	15. 62. 04
E2 reg word separator	Set E2 reg delayed E-ch word-separator mode E1 reg word separator	15. 41. 11

12. Transfer E1 to E2. If the instruction is given in the load mode with a word separator character in E2 and no word separator in E1, copy E1 BCD and generate a word mark in E2.

<u>Signal</u>	<u>Control</u>	<u>Logic</u>
Copy inv E1 WM · C-bit	E2 reg word separator (not) E1 reg word separator E-ch input mode (not) Gate console to assembly	15. 62. 06
Copy E1 BCD to E2 reg	E2 reg word separator	15. 62. 06
E2 reg full latch (on)	Set E2 reg E1 reg not word separator	15. 41. 10

13. Transfer E1 to E2. If the instruction is given in the load mode with a word separator character in E1 and E2, set E2 with E1 BCD and C-bit.

<u>Signal</u>	<u>Control</u>	<u>Logic</u>
Copy E1 BCD to E2 reg	E2 reg word separator	15. 62. 06
Copy E1 WM · C-bit	E2 reg word separator E1 reg word separator E1 reg not WM bit	15. 62. 06

14. Initiate a storage cycle in the CPU whenever E2 is full (E2 full latch on). If the command is unoverlapped, take an E-cycle and read out the BAR.

<u>Signal</u>	<u>Control</u>	<u>Logic</u>
E-cycle required	E-ch-in-process E-ch output mode (not) E-ch int end-of-transfer E1 reg full latch (off) (not) E-cycle · any last gate	12. 12. 62
(Start logic clock)	E-cycle required E-ch unovlp-in-process	11. 10. 10
E-cycle ctrl	E-cycle required 2nd clock pulse Logic gate Z (not) E-ch int end-of-transfer (not) F-cycle required	12. 12. 66

<u>Signal</u>	<u>Control</u>	<u>Logic</u>
E-cycle	E-cycle control Logic gate B or S	12. 12. 02
RO BAR	E-cycle ctrl E-ch unovlp-in-process (not) Console inhibit AR RO (not) E-ch 2nd address unovlp (not) 1st I/O cycle control	14. 71. 31

15. Initiate a storage cycle in the CPU whenever E2 is full (E2 full latch on). If the command is overlapped, take an E-cycle and read out EAR.

<u>Signal</u>	<u>Control</u>	<u>Logic</u>
E-cycle required	E-ch-in-process E-ch output mode (not) E1 reg full (not) E-ch int end-of-transfer (not) E-cycle · Any last gate	12. 12. 62
(Start logic extension clock)	E-cycle required E-ch overlap-in-process	11. 10. 20
E-cycle ctrl	E-cycle required 2nd clock pulse Logic gate Z (not) E-ch int end-of-transfer (not) F-cycle required	12. 12. 66
E-cycle	E-cycle ctrl Logic gate B or S	12. 12. 66
RO EAR	E-cycle ctrl E-ch ovlp-in-process (not) Console inhibit AR RO (not) E-ch 2nd addr ovlp	14. 71. 35

16. Gate the E2 character to the A-channel.

<u>Signal</u>	<u>Control</u>	<u>Logic</u>
Gate E2 data reg to A-ch	E-ch input mode E-cycle (not) Control reg disable	15. 38. 03

17. End the internal transfer, if the B-channel character is a group-mark with a word-mark.

<u>Signal</u>	<u>Control</u>	<u>Logic</u>
B-ch group mark · WM	B-Ch 1 · 2 · 4 · 8 · A · B · WM	15. 41. 07
E-ch end-of-record latch	E-cycle B-cy group mark · WM Logic gate F R-symbol op modifier (not) 1401 card print-in-process	13. 63. 01

18. End the internal transfer. If a wrap-around condition occurs.

<u>Signal</u>	<u>Control</u>	<u>Logic</u>
E-ch end-of-record latch	(not) E-ch last input cycle Wrap-around conditions Logic gate F or W 2nd clock pulse 2 E-cycle	13. 63. 01
E-ch int end-of-transfer	E-cycle (not) E-ch last input cycle Wrap-around conditions Logic gate F or W 2nd clock pulse 2	13. 63. 01

19. End internal transfer. If the external end of transfer occurs before the internal end of transfer, take an extra input cycle to search for the group-mark with a word-mark, and end internal transfer.

<u>Signal</u>	<u>Control</u>	<u>Logic</u>
E-ch last input cycle	E-ch ext end-of-transfer (not) E1 reg full (not) E2 reg full E-cycle E-ch input mode Logic gate C or T	13. 63. 02
E-ch int end-of-transfer	E-ch last input cycle Logic gate F or W	13. 63. 01

20. Develop assembly controls. If the asterisk insert switch is ON and the A-channel character is invalid, set an asterisk.

<u>Signal</u>	<u>Control</u>	<u>Logic</u>
In cy GM·WM ctrl	B-ch not group mark·WM Input cycle (not) Any last input cycle	15. 49. 02
A-ch invalid	(Invalid bit configuration)	18. 11. 03
Set asterisk	In cy GM·WM ctrl A-ch invalid (not) 1 asterisk ins console sw (off)	15. 49. 06

21. Develop assembly controls. If the asterisk insert switch is OFF, and if the A-channel character is invalid, stop the clock.

<u>Signal</u>	<u>Control</u>	<u>Logic</u>
Master error	A-channel VC error Logic gate A or R	18. 14. 08
Stop latch	Master error	12. 15. 04

22. Develop assembly controls. If the the instruction is given in the load mode with a valid A-channel character, use the A-channel numerical bits, word-mark bit, and zone bits.

<u>Signal</u>	<u>Control</u>	<u>Logic</u>
A-ch valid + ast switch (off)	A-ch valid or Asterisk ins console sw (off)	15. 49. 02
E-ch select odd-parity unit	E-ch select unit 1	13. 60. 03
Odd-parity cycle	E-ch select odd-parity unit E-cycle	13. 60. 02
Use A-ch zones	In cy GM·WM ctrl A-cy valid + ast switch (off) Odd-parity cycle	15. 49. 03
Use A-ch nu	In cy GM·WM ctrl A-ch valid + ast switch (off) Odd-parity cycle	15. 49. 03
Use A-ch WM	In cy GM·WM ctrl A-ch valid + ast switch (off) Load cycle	15. 49. 02

23. Develop assembly controls. If the instruction is given in a move mode with a valid A-channel character, use the A-channel numerical bits and zone bits and the B-channel word-mark bit.

<u>Signal</u>	<u>Control</u>	<u>Logic</u>
Use A-ch zones	In cy GM·WM ctrl A-ch valid + ast switch (off) Odd-parity cycle	15. 49. 03
Use A-ch nu	In cy GM·WM ctrl A-ch valid + ast switch (off) Odd-parity cycle	15. 49. 03
Use B-ch WM	Move cycle Input cycle	15. 49. 04

24. Develop assembly controls. If the internal end-of-transfer is detected, use the full B-channel character.

<u>Signal</u>	<u>Control</u>	<u>Logic</u>
I/O grp mk·WM stop ctrl	E-ch ovlp-in-process E-cycle or E-ch unovlp-in-process E-cycle R-symbol op modifier	13. 65. 04

<u>Signal</u>	<u>Control</u>	<u>Logic</u>
Use Full B-ch	I/O grp mk-WM stop ctrl B-ch group mark · WM Input cycle (not) 1401 card print in-process	15. 49. 05

25. Develop assembly controls. If the CPU is in the last input cycle, use the full B-channel character.

<u>Signal</u>	<u>Control</u>	<u>Logic</u>
Any last input cycle	E-ch last input cycle	13. 63. 02
Any last in cycle not 1401	Any last input cycle (not) 1401 mode	15. 49. 05
Use B-ch nu	Any last in cycle not 1401	15. 49. 05
Use B-ch zones	Any last in cycle not 1401	15. 49. 05
Use B-ch WM	Any last in cycle not 1401	15. 49. 04

26. Reset the E2 full latch after each input cycle.

<u>Signal</u>	<u>Control</u>	<u>Logic</u>
Reset E2 full latch	Any last gate E-ch cycle In cy GM-WM ctrl 2nd clock pulse (inverted)	15. 62. 05

27. After both internal and external transfers end, develop a sample pulse for setting the E-channel-status indicators.

<u>Signal</u>	<u>Control</u>	<u>Logic</u>
E-ch int end-of-tfr delayed	Logic gate Z E-ch int end-of-transfer	13. 65. 05
E-ch status sample B	E-ch ext end-of-transfer E-ch int end-of-trf delayed Logic gate Z 2nd clock pulse 1 (inverted)	13. 65. 05

28. Set the E-channel wrong-length-record status indicator, if the record length was incorrect.

<u>Signal</u>	<u>Control</u>	<u>Logic</u>
E-ch wrong-length record	E-ch status sample B E-ch strobe trigger or E2 reg full or (not) E-ch end-of-record latch	13. 63. 03

29. Set the E-channel data-check status indicator, if any error occurred during the transfer.

<u>Signal</u>	<u>Control</u>	<u>Logic</u>
Errors 2	Pwr clock error or Ring error to error latch or Parity 1 or Parity 2	52. 12. 01
PT errors	Errors 2 PT rd scan or Parity error to adapter or PT, ready-error	55. 10. 06
14C5-A ext error status	PT rd op PT errors	55. 10. 04
Buffer error	14C5-A ext error status	51. 40. 21
E-ch check bus	Buffer error E-ch select any buffer	12. 62. 03
E-ch check	E-ch check bus E-ch status sample B	12. 62. 04

30. Continue with the stored program. If the command is unoverlapped, set last execute cycle and reset the unoverlap-in-process latch.

<u>Signal</u>	<u>Control</u>	<u>Logic</u>
Last execute cycle I/O	E-ch unovlp-in-process (not) 1401 card or print op code E-ch status sample B	13. 65. 07
E-ch second sample B	E-ch status sample 3 2nd clock pulse (inverted)	13. 65. 05
E-ch status sample B-delay	E-ch second sample B 2nd clock pulse (inverted)	13. 65. 05
(Reset) E-ch unovlp-in-process	E-ch status sample B-delay	13. 60. 04

31. Continue with the stored program. If the command is overlapped, reset the overlap-in-process latch.

<u>Signal</u>	<u>Control</u>	<u>Logic</u>
(Reset) E-ch ovlp-in-process	E-ch status sample B-delay	13. 60. 04





### 3.4.00 CHECKING FEATURES

#### 3.4.01 Parity Check (Logic 51.18.01)

Information is stored in the integrated-synchronizer core-storage unit in odd-parity form. That is, every character must contain an odd number of bits. The bits that are set up at the inhibit driver inputs are switched to two parity check triggers (Figure 3.4-1). Both triggers are reset OFF at the beginning of each memory cycle. Binary trigger inputs allow each trigger to flip ON and OFF with successive set pulses. The C, A, 4, and 1-bits are switched to the upper trigger at 6, 7, 8, and 9-times, respectively. The B, 8, and 2-bits are switched to the lower trigger at 6, 7, and 8-times, respectively. On a serial device, the 8th or control bit must be considered in the parity check of each character. This 8th bit is accounted for by an entry to the lower trigger at 9 time. If there is an even number of bits at the inhibit-driver inputs, the two triggers are either both ON or both OFF at ten time when the error sample occurs. An odd number of bits causes the triggers to be opposite (not both on or off) when they are tested at ten time, in order to prevent an error.

#### 3.4.02 Ring Check (Logics 51.31.02, 51.32.02)

A ring-check circuit maintains a constant check on the proper progression of the units and tens-ring triggers. For each ring, there is a check trigger driven by the corresponding-ring drive pulse. Each check trigger has a binary input; therefore, each incoming drive pulse flips the trigger to the opposite state. The outputs of the check triggers are compared with the outputs of the corresponding ring triggers. The check triggers should be ON when the even-numbered ring triggers are ON, and OFF when the odd numbered triggers are ON. Any deviation from this indicates incorrect ring progression and produces a ring check (Figure 3.4-2).

The ring check, in addition to resetting all ring triggers, sets the check latch of the unit that is being addressed when the error is detected. For example, if a ring check occurs during a punch scan, the punch-check latch is set.

#### 3.4.03 Clock Check (Logic 51.30.01)

The clock check circuit detects any incorrect progress of the clock triggers. Clock pulses drive a binary-input check trigger. Because the memory cycle has an odd number of pulses, the check trigger resets ON at zero time of each memory cycle. The check trigger(reset ON at zero time) is ON at 0, 2, 4, 6, 8, and 10 times, and OFF at 1, 3, 5, 7, and 9 times. The outputs of the even-numbered clock triggers are compared with the on output of the check trigger. The outputs of the odd-numbered clock triggers are compared with the OFF output of the check trigger (Figure 3.4-3).

If a clock check occurs, the clock triggers are reset and allowed to restart. In addition, the check latch of the unit that is being addressed is set to indicate a data error.

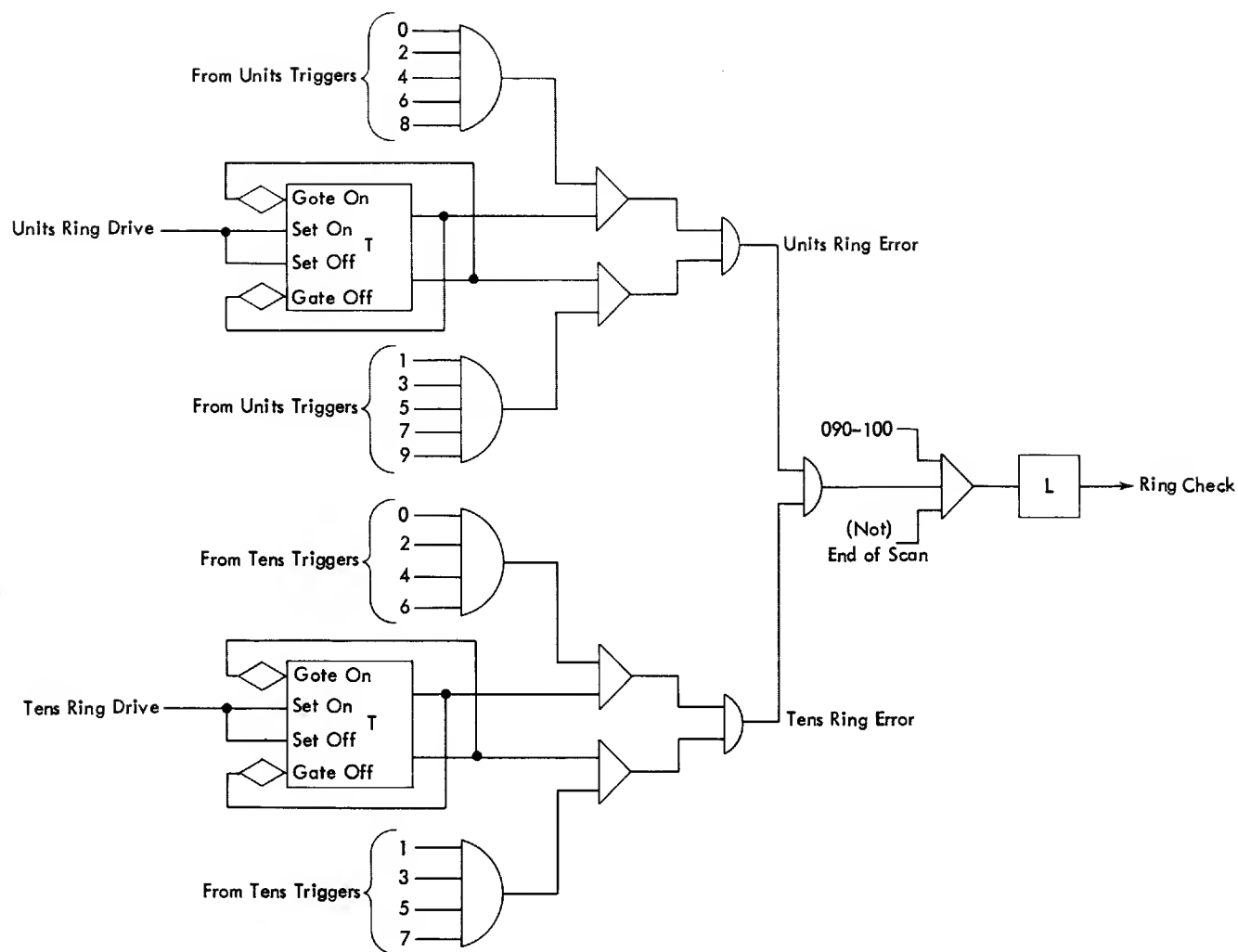


Figure 3.4-2 Ring Check

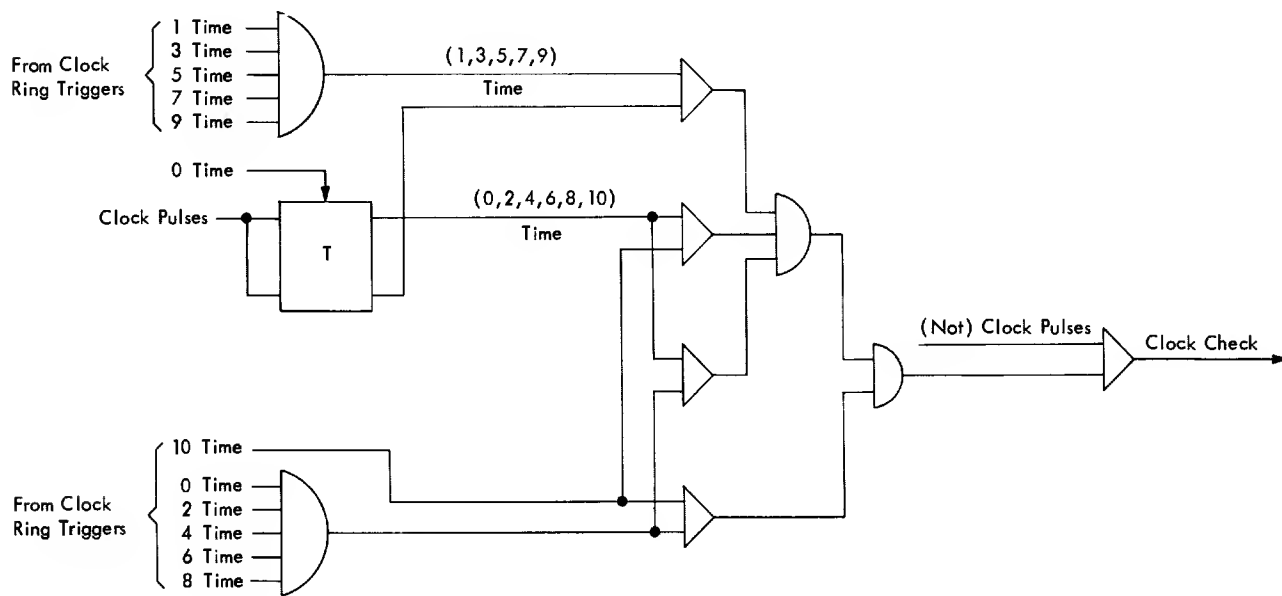


Figure 3.4-3 Clock Check

#### 3.4.04 Hole-Count Check

##### Operation

To insure integrity of card input-output data, the IBM 1410 uses a hole-count check circuit that consists of eight 80-core planes and their associated circuitry. Four of these are used by the card reader and four are used by the card punch. These planes are labeled Upper X and Upper Y, and Lower X and Lower Y for each machine.

Hole counting by the card reader is a comparison of the number of holes that are counted in each card column at the two read stations. Essentially, the machine counts the holes in each column of a card when it is read at the read-check station. It then subtracts the holes in each column from the same card when it reaches the read station. The result should always be zero, because the same number of holes should be read at both read stations.

There are two hole-count check cores that are associated with each read-one and read-two-row bit-core position.

The upper Y (UY) and lower Y (LY) cores do the counting at the read-check station. The upper X (UX) and lower X (LX) cores count the information from the read station.

At the beginning of the card feed cycle, all cores are in the 0-status. During every read scan, all 80 positions of the check cores are successively addressed and are set with the required hole-count information.

The first hole in each card column that is sensed at the read-check station causes both Y-cores associated with that column to flip. Thereafter, the UY-core remains unchanged, while the LY-core status is reversed with each successive hole that is read in its respective column. At the end of the first card-feed cycle, the reader Y-planes define two conditions for each card column: whether a hole was read (UY), and whether the total count was odd or even (LY). At reader-9-time during the next card-feed cycle, the contents of all 80 UY and LY cores are transferred to the X-plane circuits. As the card passes the read brushes, the punched card data combines with the transferred Y-plane data. The result, at reader-12-time, is a 0-status of all cores in the X-planes.

Logic (Figures 3.4-4 and 3.4-5)

The memory cycle, that contains both read and write times, controls the status of the check cores. As each hole-count check position is addressed, the cores that are associated with that position read out in order to set data registers. The core-controlling circuit analyzes the read data along with incoming data from row-bit storage, and determines whether the core should be set or inhibited on the write portion of the cycle. This takes place 80 times for each of the 12 rows of information in a card.

Circuits (Figures 3.4-6 and 3.4-7)

The card with an L (11-3) punched in card column 2 is read in and checked by the card reader. The following discussion is based on a no-error condition.

CARD FEED CYCLE	READER TIME	HOLE READ	CIRCUIT OBJECTIVES AND REASONS	CORE STATUS AT END OF READ SCAN			
				UY	LY	UX	LX
1 ↓	9 Through 4	NO	Inhibit UY - No UY Or Row Bit Data Inhibit LY - No UY or Row Bit Data	0	0		
	3	YES	Set UY - Row Bit Data Set LY - Row Bit Data, Na LY Data	1	1		
	2 Through 0	NO	Regen UY - UY Data Regen LY - LY Data - No Row Bit Data	1	1		
	11	YES	Regen UY - UY Data Inhibit LY - LY Data - Row Bit Data	1	0		
	12	NO	Regen UY - UY Data Inhibit LY - No LY Data - No Row Bit Data	1	0		
2 ↓	9	NO	Set UX - Uy Data - No Row Bit Data Inhibit LX - Na LY Data - Na Row Bit Data			1	0
	8 Through 4	NO	Regen UX - UX Data - No Row Bit Data Inhibit LX - No LY Data - Na Row Bit Data			1	0
	3	YES	Inhibit UX - UX Data - Row Bit Data Set LX - Na LY Data - Row Bit Data			0	1
	2 Through 0	NO	Inhibit UX - Na UX Data - Na Row Bit Data Regen LX - LX Data - Na Row Bit Data			0	1
	11	YES	Inhibit UX - No UX Data - Row Bit Data Inhibit LX - LX Data - Row Bit Data			0	0
	12	NO	Inhibit UX - No UX Data - Na Row Bit Data Inhibit LX - No LX Data - No Row Bit Data Prevent Error - No Data To Cores			0	0

Figure 3.4-4 Hole-Count Check Example

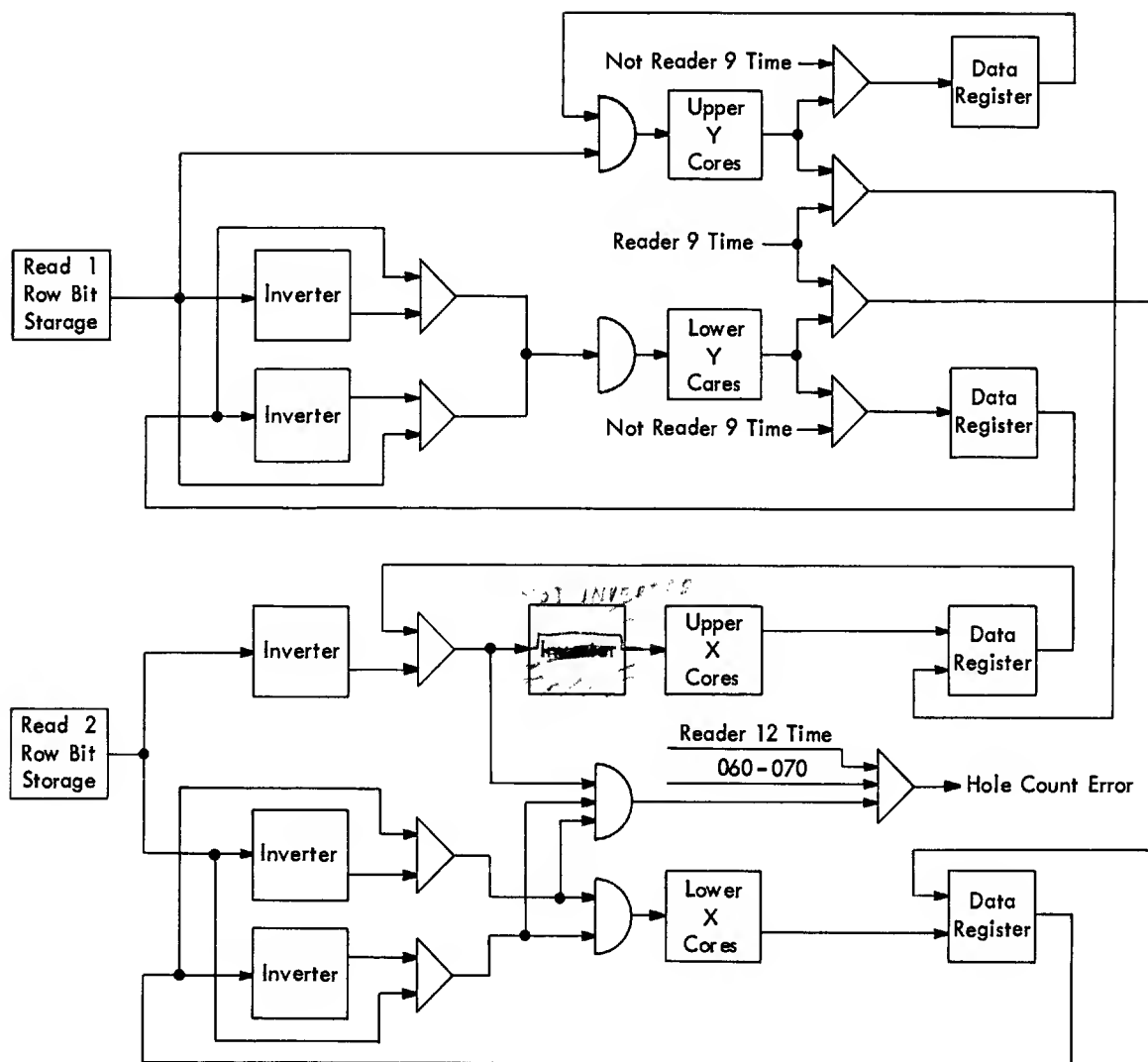


Figure 3.4-5 Hole-Count Check Logic

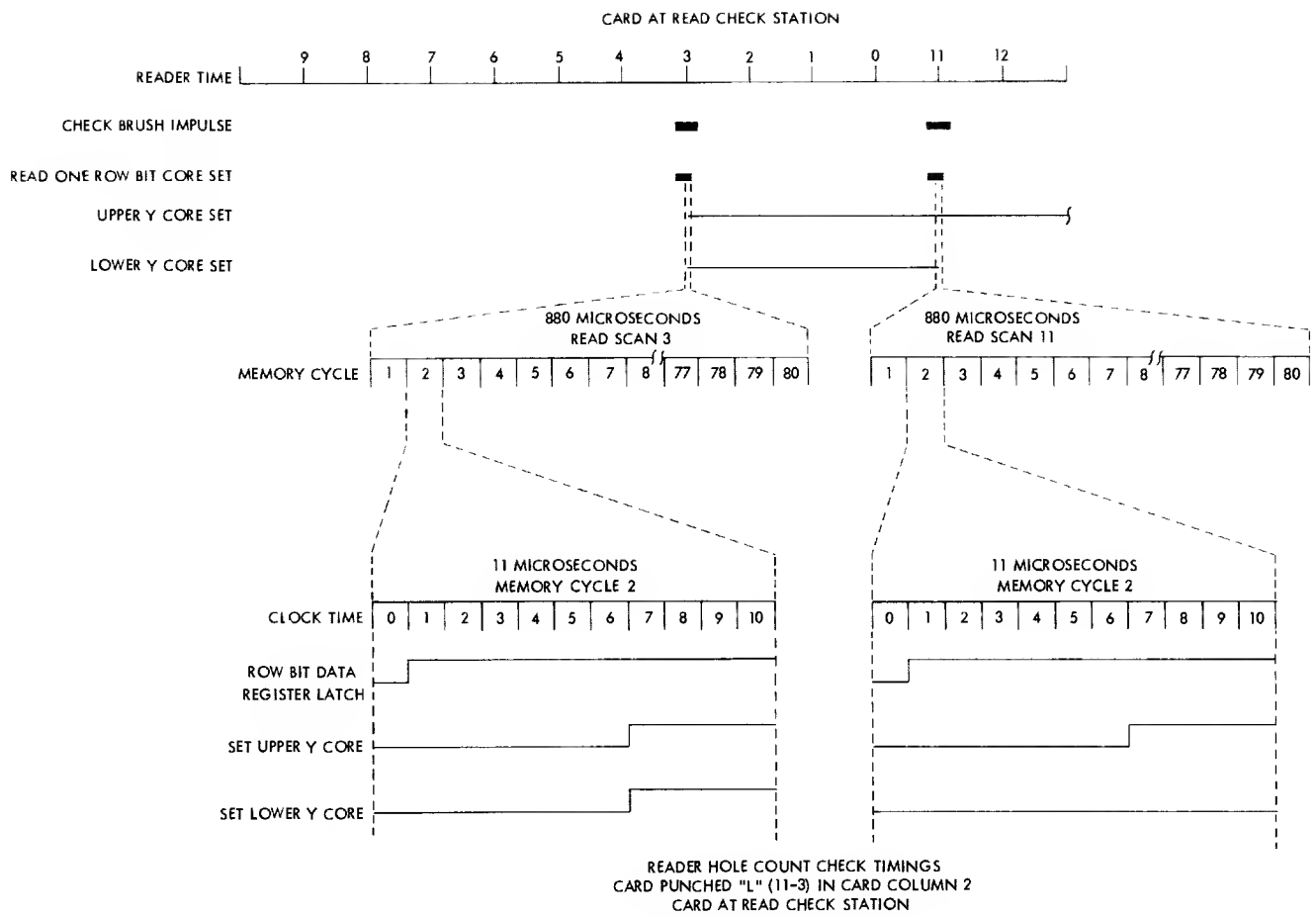


Figure 3.4-6 Reader Hole-Count Check Timings

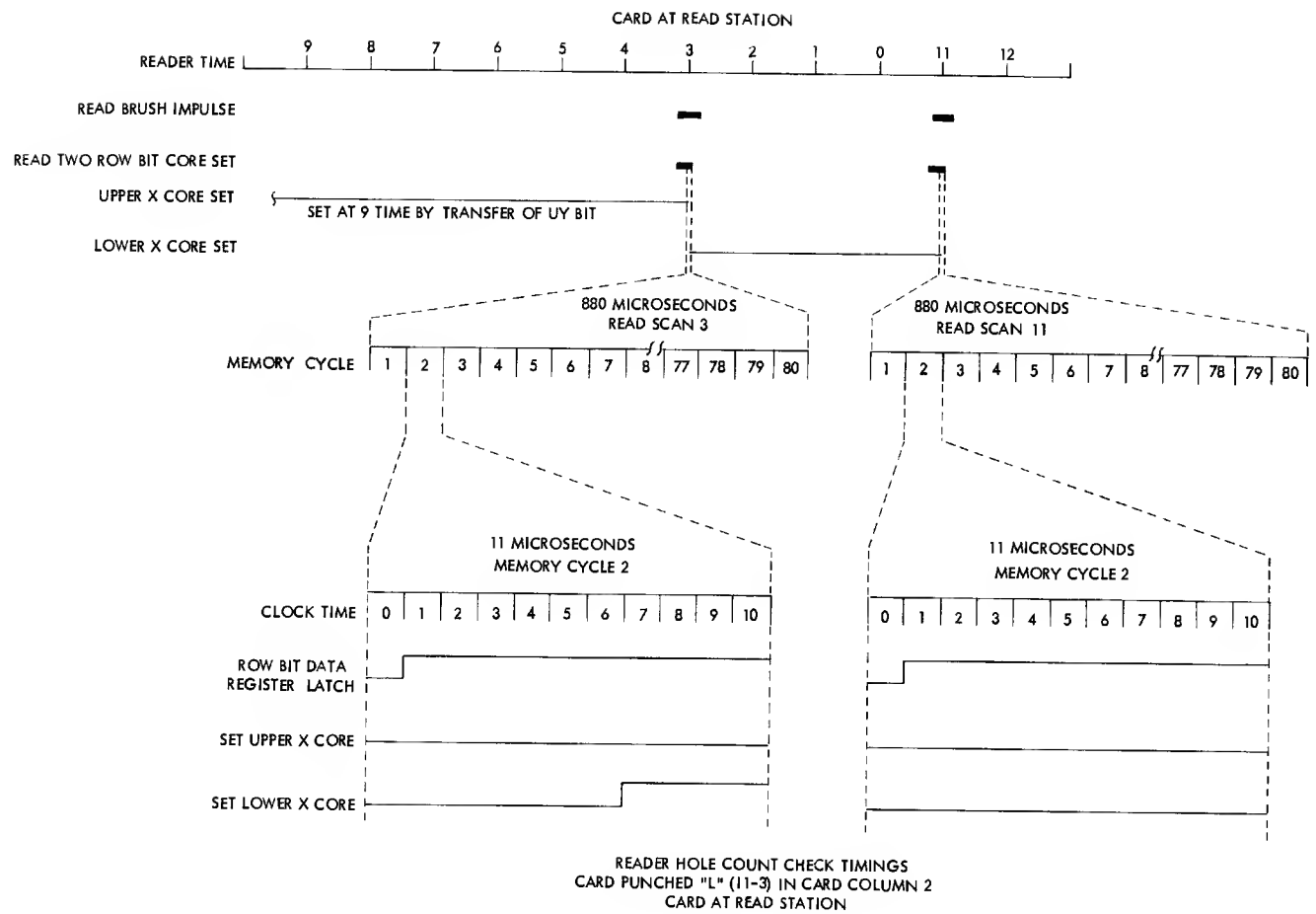


Figure 3.4-7 Reader Hole-Count Check Timings

1. At reader-9-through-4-time, no holes are read from the card. At reader-3-time, during the second memory cycle, the three punch from card column two is read out of read-one-row bit-storage (position two), and is switched to set both UY and LY cores for that position. Both cores must be set, because this is the first hole that is read in that column.

<u>Signal</u>	<u>Control</u>	<u>Logic</u>
Data reg out 1st rd	Read pulse 010-040	51. 10. 01
Hole-ct to inh drvr up Y	Data reg out 1st rd	51. 14. 01
Hole-ct to inh drvr lwr Y	Data reg out 1st rd (not) Data reg out LY	51. 14. 01

2. During the second memory cycle of read-scan 11, position two of UY is regenerated, and the input to LY-core for position two is blocked, because this is the second hole that is read in that column.

<u>Signal</u>	<u>Control</u>	<u>Logic</u>
Data reg out 1st rd	Rd pulse 010-040	51. 10. 01
Hole-ct to inh drvr UY	Data reg out UY or Data reg out 1st rd	51. 14. 01
(not) Hole-ct to inh drvr LY	Data reg out LY	51. 14. 01

3. At reader-9-time of the second card-feed cycle, the contents of all Y-cores are transferred to the respective X-cores on successive memory cycles. Before setting the X-cores, the Y-data is combined with any read-two-row bit-data that is present. In this case, however, no holes are read at 9-time, so that there is no read-two bit-data.

<u>Signal</u>	<u>Control</u>	<u>Logic</u>
Y A Trf	Rd scan Read-9-time Read pulse 010-040	52. 10. 05

4. At 3-time of the second card-feed cycle, the read-two-row bit-data from card column two combines with position-two X-core-data to prevent setting of position-two-UX, and to set position-two-LX.

<u>Signal</u>	<u>Control</u>	<u>Logic</u>
Data reg out 2nd rd	Rd pulse 010-040	51. 10. 01
(not) Hole-ct to inh drvr up X	Data reg out up X Data reg out 2nd rd	51. 14. 01
Hole-ct to inh drvr lwr X	Data reg out lwr X (not) Data reg out 2nd rd	51. 14. 01



5. At reader-11-time of the second card-feed cycle, the read-two-row bit-data combines with X-data to block the setting of both UX and LX cores for position two.

<u>Signal</u>	<u>Control</u>	<u>Logic</u>
Data reg out 2nd rd	Rd pulse 010-040	51. 10. 01
(not) Hole-ct to inh drvr up X	(not) Data reg out up X	51. 14. 01
(not) Hole-ct to inh drvr lwr X	Data reg out lwr X Data reg out 2nd rd	51. 14. 01

6. To detect a hole-count error, the output of reader X-plane switching is tested at 12-time in order to determine whether any cores are to be set (any X-core set after 12-time indicates a hole-count error).

<u>Signal</u>	<u>Control</u>	<u>Logic</u>
Hole-ct chk gate	Rd req rd scan Reader 12 time	52. 10. 06
(not) Hole-ct err	(No data from chk cores)	51. 14. 01

The hole-count check circuit for the card punch is similar to that for the card reader. The same switching circuits and data registers are used by both reader and punch, because only one scan can occur at any given time. The differences lie in the sources of hole-count input data, and the times when the scans occur.

The punch, instead of having two reading stations, has one punch station and one read station (Figure 3.4-8). BCD data from the punch buffer is decoded into the IBM card-code for punching. At the same time, the BCD data feeds through a bit generator (Logic 51.13.01) to translate the information into row-bit form. The punch data, now in row-bit form, becomes one source of information for the hole-count check circuit; it is comparable to read-one-row bit data from the card reader. The other source of information that is used for the count comparison is the row-bit data from the punch check brushes. This is comparable to card reader read-two-row bit-data, and is read exactly one card feed cycle after it is punched. The punch hole-count circuit (Figure 3.4.8), detects any error in punch decoding, or any failure to punch the required holes.

In order to punch holes in a card, the punch magnets must be set up before actual punching time. This means that the first scan takes place before 12-time of the punch cycle. At this time, the 12-row of the card at the punch-check station has not been read and therefore cannot enter into the hole-count check circuits. The 12-row data are scanned out of row-bit storage during the second punch scan, or while the 11-row is being set up to punch. Consequently, hole-count data from the punch-check brushes are always one cycle-point late with respect to the punch hole-count data. At 9-time, the 9-row is set up in the punch magnets, and the 8-row card data is scanned out of row-bit storage to the hole-count check circuit. In order to complete the hole-count check, an after-9-cam initiates an extra or thirteenth scan. This additional scan allows the 9-row card data to be scanned out to the hole-count check circuit to complete the check.

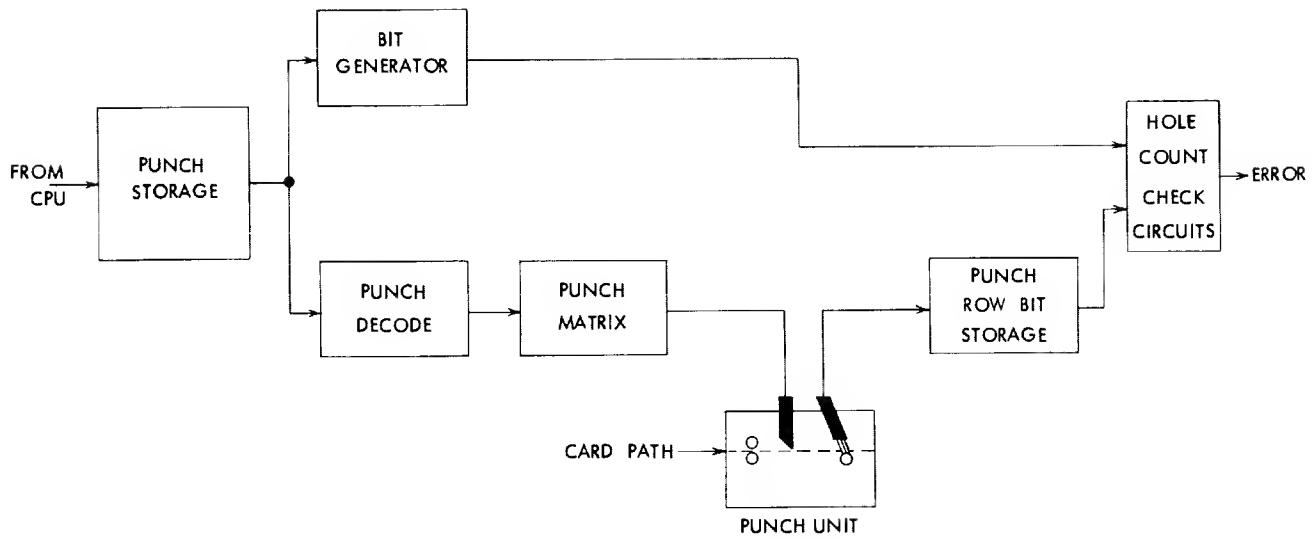


Figure 3.4-8 Punch Hole-Count Check

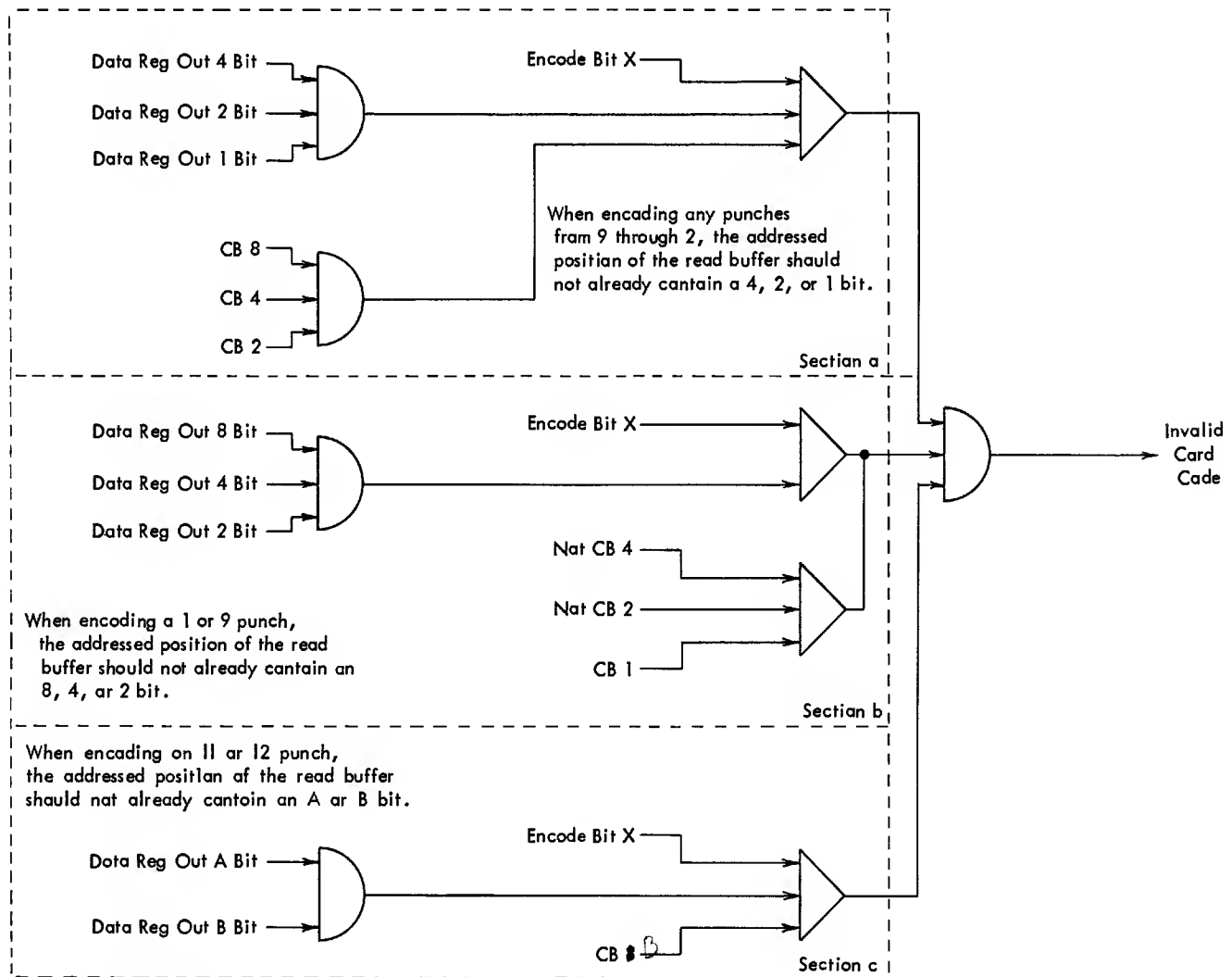


Figure 3.4-9 Reader Validity Check

### 3.4.05 Reader Validity Check

#### Operation

An IBM card, with its 12 punching positions per column, can contain a great number of punch combinations. Of these, only 64 are recognized as representing valid characters. All other combinations are considered invalid. The reader validity-check circuit analyzes each character as it is being coded. The sensing of any incorrect character coding results in an invalid-card code signal.

During each read scan, information that is scanned out of row-bit storage combines with information that is scanned out of the read buffer in order to determine the validity of all characters while they are being developed. This check must be performed during every memory cycle of each read scan, because the machine reads in a parallel manner by row and serially by card hole.

The reader validity-check circuit consists of three sections. Each can detect a different type of coding error (Figure 3.4-9). The conditions that each section must meet determine the validity of each character, as follows:

1. Section A. When coding any punches from 9 through 2, the addressed position of the read buffer should not already contain a 4, 2, or 1-bit.
2. Section B. When coding a 1- or 9-punch, the addressed position of the read buffer should not already contain an 8, 4, or 2-bit.
3. Section C. When coding an 11- or 12-punch, the addressed position of the read buffer should not already contain an A- or B-bit.

Any invalid-card code is detected by one of the three sections. This results in an invalid-card code signal.

Read-request, read-scan and bit X-signals develop the encode-bit X signal. Bit X represents read-two-row bit-data.

#### Circuits

A card column that contains a 9- and an 8-punch is checked to indicate a validity error.

The 9-punch from row-bit storage feeds to the validity-check circuit and is called bit X. No error is indicated because there are no bits in the addressed buffer position.

<u>Signal</u>	<u>Control</u>	<u>Logic</u>
Encode	Rd req. rd scan	52, 10, 05
Encode bit X	Encode	51, 12, 01

At reader-8-time, the 8 from row-bit storage (bit X) combines with the 8-bit and 1-bit from the addressed buffer position to produce an invalid-card core signal.

<u>Signal</u>	<u>Control</u>	<u>Logic</u>
Invalid-card code	Bit X, data reg out 1-bit and read CB8.	S1. 12. 01

## 4.0.00 PRINT STORAGE

### 4.1.00 PRINTER CONTROL (Figure 4.1-1)

This section explains the circuitry in the IBM 1414 Input-Output Synchronizer that is necessary to control printing and carriage operations. For information on principles of operation of the printer and carriage, refer to IBM 1403 Customer Engineering Manual of Instruction, 225-6492.

#### 4.1.01 Printer Core-Storage Unit

Print storage is an 11-plane, 132-position, core-storage unit. Information to be printed is placed in print storage during a read-in operation. When read-in is complete, a scan-out operation reads the stored print-information at the speed and in the order that the IBM 1403 Printer requires. Seven core planes (C, B, A, 8, 4, 2, 1) store characters in the modified BCD code. The remaining four planes (hammer-check, print-line-complete, equal-compare, and hammer-fire row bits) detect print errors.

Each core plane contains ten rows of 16 cores, for a total of 160 cores per plane. Of the 160 total positions only 100 or 132 are used, depending on the print positions that the printer has available. Each print position is represented by 11 cores, one in each plane.

Print storage uses the integrated-synchronizer clock as the source of pulses and gates.

On a read-in operation, print information is translated to the modified BCD code that the printer requires. The translated characters are placed in print storage in the seven data-bit planes of the addressed position. An odd-bit parity check is made of the modified BCD characters. Each character is also tested to determine whether it can be printed. The CPU recognizes 64 valid characters, but the printer can print only 51 characters (on 48 type characters).

Scan-out occurs after read-in is complete. As each position is addressed, the information in the cores reads out to set the data register latches. If the character that reads out corresponds to the type character that is aligned at the print position, an Equal Compare results and the hammer fires. After the compare operation, the character is placed back in storage until the next print scan.

The core array is arranged physically and electrically in such a way that information from the CPU reads into sequential print storage positions (1, 2, 3, 4, etc). When this information is to be printed, it is scanned out in address increments of three (1, 4, 7, 10, etc), as the printer requires.

#### Addressing Rings

Three rings control core-storage addressing (Figure 4.1-2). The rings are called the threes, fives, and tens rings from the number of positions they contain. The

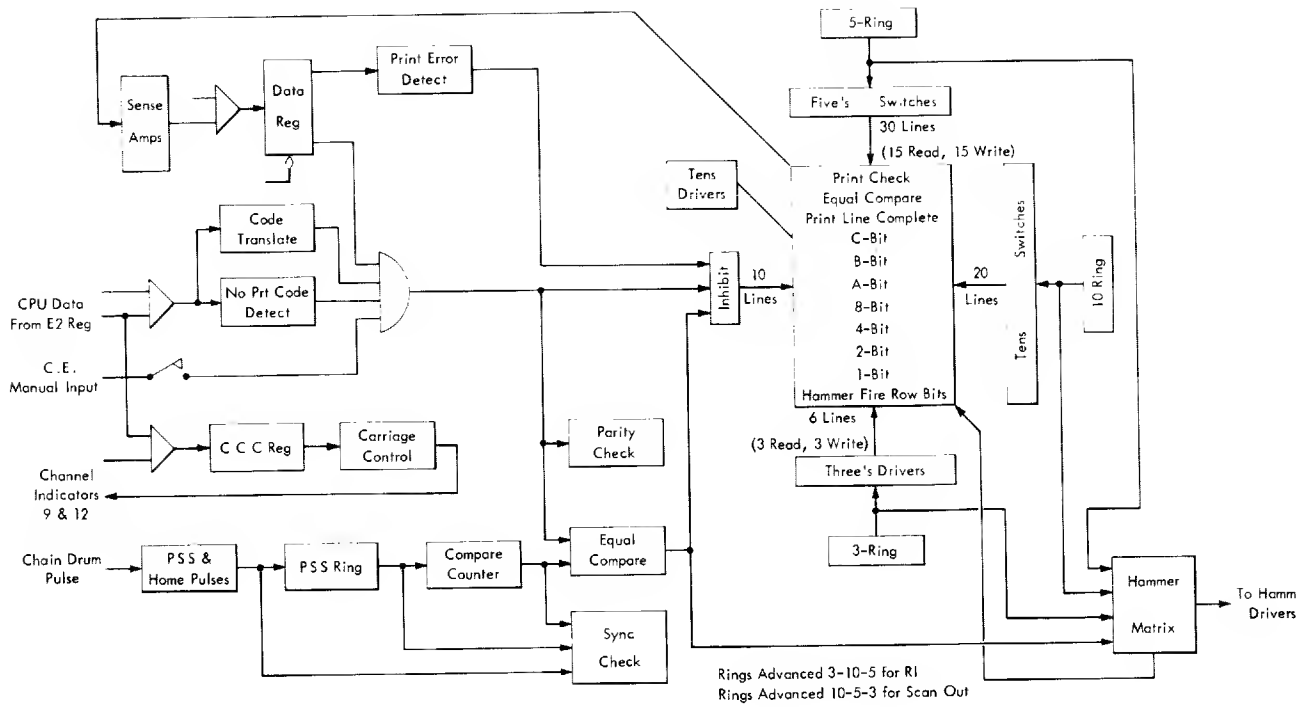


Figure 4.1-1 IBM 1414 Print Storage

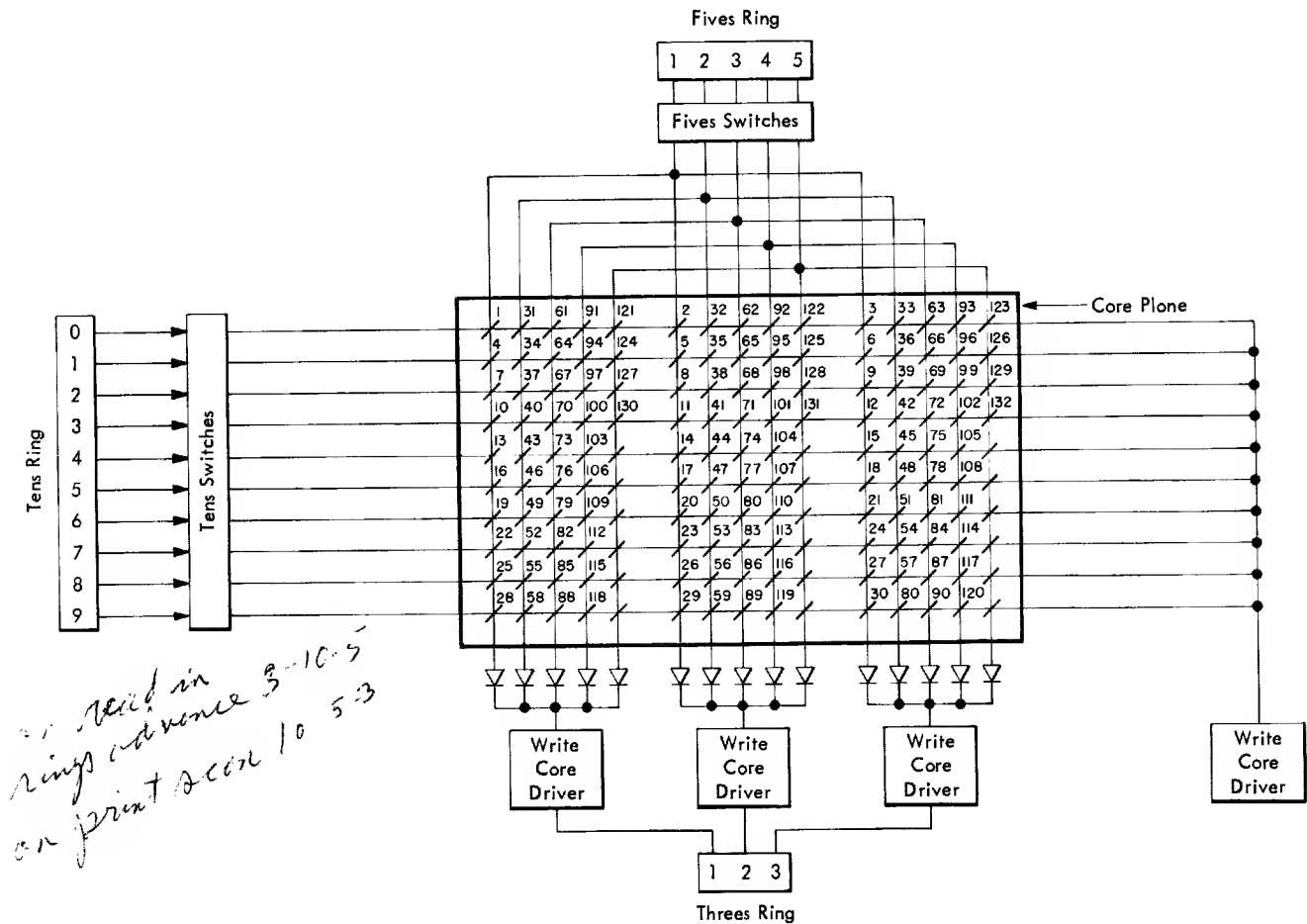


Figure 4.1-2 Print Storage Addressing Showing Write Drivers

tens ring selects a row of cores in one direction. The threes and fives rings together select a row of cores in the other direction. The coincidence of the two selected lines produces a selected-character position.

For example: assume that core position 1 is to be selected. The 0-position of the tens ring selects the tens row that contains core-position 1. The first position of the fives ring selects three lines in the other direction. Of the three lines, the core driver actuates only one. The output of the threes ring selects the correct core driver.

When core position 1 is selected, position 2 can be addressed by advancing the threes ring to the next position. When the threes ring advances to position 3, it selects core position 3. The next pulse to the threes ring causes position 1 to come back ON. However, going from position 3 to position 1 develops a threes-ring carry that, when fed to the tens ring, causes the tens ring position 1 to come ON. With position 1 ON in all three rings, core position 4 is selected. When the tens ring carries, the fives ring advances. Therefore, on a print-storage read-in, the rings advance in 3-10-5 order.

Scan-out of print information to the printer requires that every third character be read out to correspond with every third hammer in the IBM 1403. Reading out in address increments of three occurs when the rings advance in a 10-5-3 fashion. That is, the tens ring advances every time a character is to be read out. The fives ring advances on a tens ring carry. The threes ring advances on a carry from the fives ring.

#### Drive Scheme

The core array uses separate read and write core drivers. Because of the separate drivers, separate lines must be used for reading and writing. Figure 3.1-4 identifies the six lines through each core. The read and write windings pass through the cores in opposite directions. A row-bit core is also shown.

Eight current drivers supply current through the core windings for reading and writing (Figure 4.1-3). There is one read and one write driver for the ten (T0 through T9) tens switches. The fives switches require three read- and three write-current drivers (Figure 4.1-4).

Figure 4.1-5 shows the basic timings of the print-storage circuit. At the beginning of the cycle, the ring-advance pulse addresses a particular position. The read pulse that is supplied by the core drivers flips the cores that were set to induce a current into the sense winding. The sense-winding current is amplified to set a data register latch. Later in the memory cycle, the write pulse sets the proper cores. The inhibit pulse for the addressed cores that are not to be set cancels the effect of the write pulse.

#### Read-in

Information is gated to the print-storage input on a read-in operation. The information is first translated to:





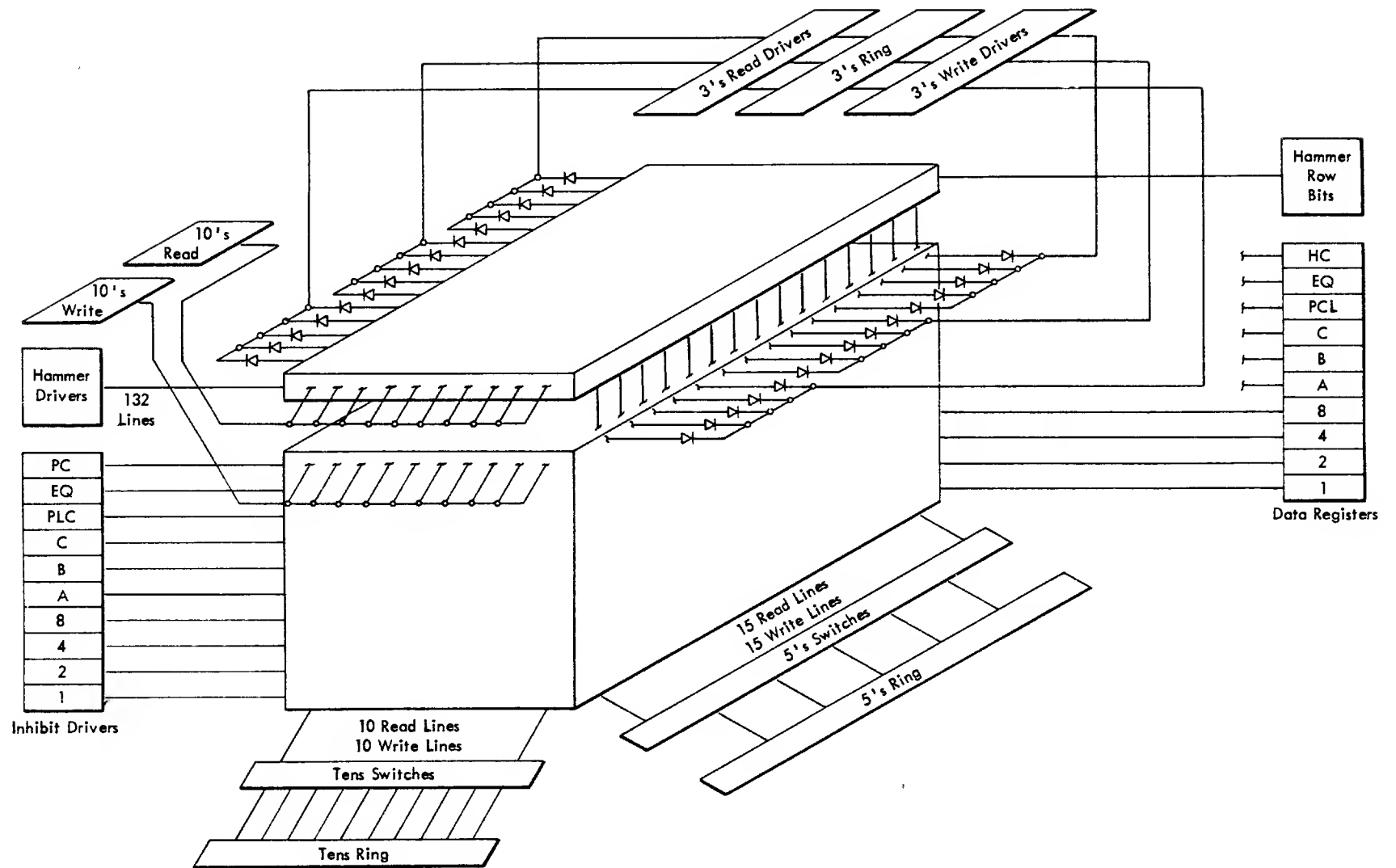


Figure 4.1-4 Print Storage Array

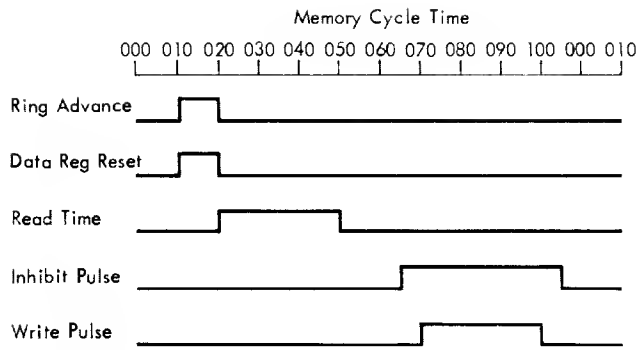


Figure 4.1-5 Print Storage Timings

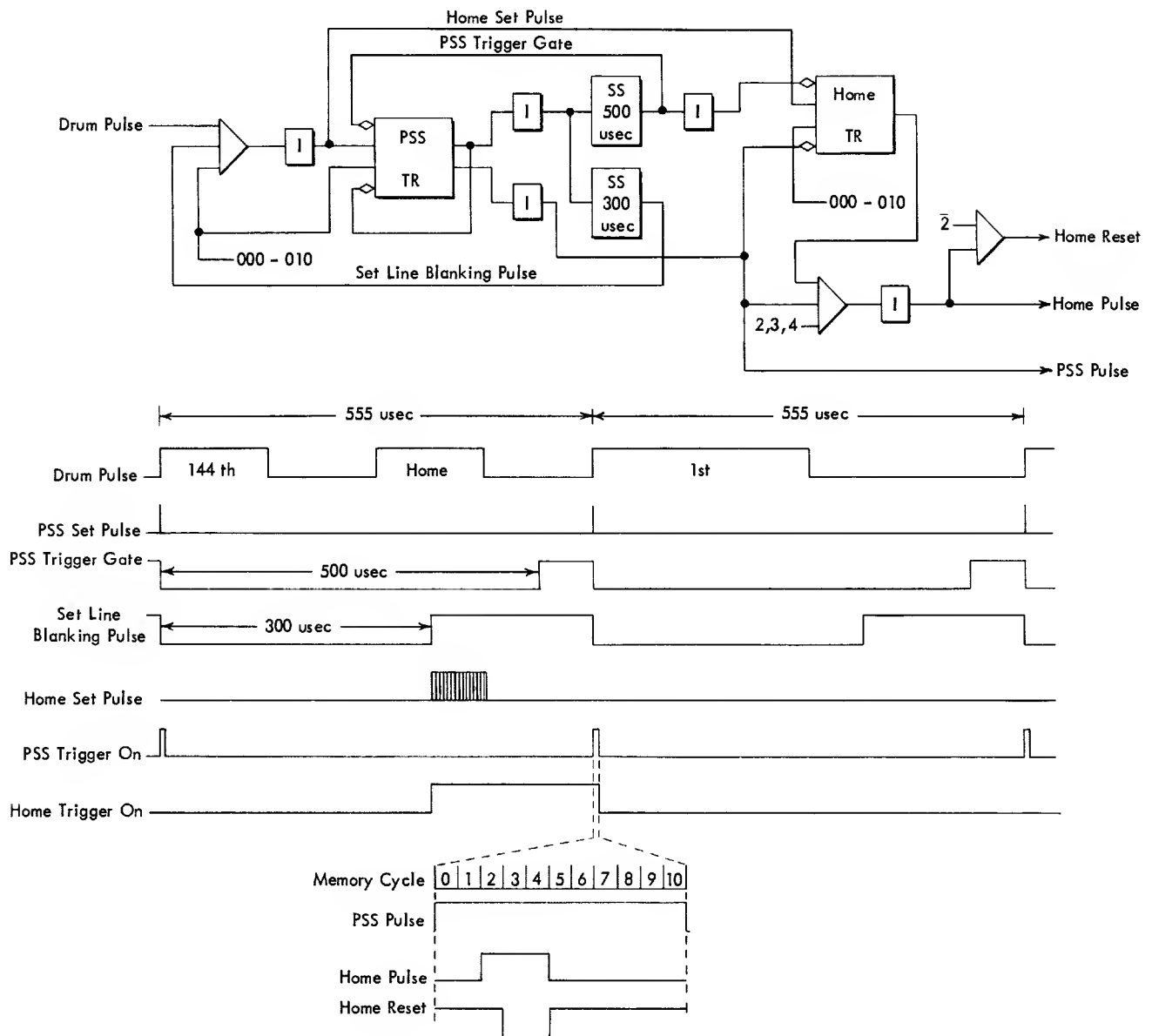


Figure 4.1-6 PSS, Home, and Home Reset Pulses

1. develop the modified BCD code that the printer requires.
2. determine whether the character is printable.

One character in the modified BCD code is placed in print storage on each 11-microsecond memory cycle. Each character that is defined as being printable sets the core for that position in the print-line complete plane. The information in this plane is used to detect errors during scan-out. The modified BCD characters are checked for odd parity. Only eight of the eleven planes can be set during read-in. The other three are used for checking during scan out.

On a read-in operation the threes ring advances at the beginning of every memory cycle. The tens ring advances at the beginning of every third cycle (threes-ring carry). The fives ring advances every 30th cycle (tens-ring carry). Each ring is gated to function as a closed ring until position 132 is reached (Logic 53. 23. 01-02). Position 132 stops the rings from advancing and gates the home trigger ON.

With machines that have only 100 print positions, all 132 are read in and scanned out. On Logic 53.11.05, C-bits are inserted during read-in in all positions after print position 100, in order to prevent parity errors while sensing out.

#### Scan-Out

During a scan-out, characters are read out in address increments of three on successive memory cycles. All eleven cores read out of each addressed position during the read portion of the memory cycle. The six data bits feed to the equal-compare circuitry to determine whether the correct type-character is aligned to the addressed print position. The data bits that are read out during the read portion of each memory cycle are regenerated during the write portion of the cycle, so that the characters will be available on successive print scans. The error-detection cores (equal-compare, hammer-fire, and hammer-check) are set up on each memory cycle. They are not tested until one print-scan later. That is, print-scan 20 is not tested for error until print-scan 21, etc. In order to test print-scan 48 (the last print-scan) for errors, an additional or 49th scan must be taken. For conditions that are recognized as errors see section 4. 3. 03 Print-Error Detector.

On a scan-out operation, the tens ring advances at the beginning of each memory cycle. The fives ring advances every ten cycles (tens-ring carry). The threes ring advances every 44 cycles (end of subscan).

Three ring-advance triggers provide the pulses necessary to control the advance of the three rings (Logic 53. 23. 02).

#### 4. 1. 02 Type Synchronization

Timing drum pulses are amplified and used to develop the print subscan and home pulses (Figure 4. 1-6). The print subscan (PSS) pulse occurs whenever a type aligns at one of the first three printing positions. The home pulse occurs when the type character 1 aligns at print position 1.

The PSS pulse drives a three-position closed PSS ring. The outputs of the ring triggers (called PSS-1, PSS-2, and PSS-3 pulses) identify the subscan that is taking place.

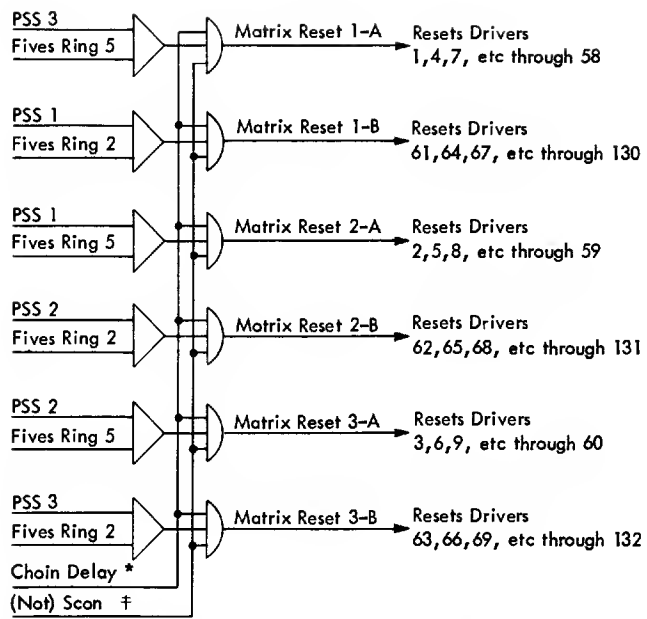
There are 49 print-scans within any print line. A six-position print-scan counter advances at the beginning of each print-scan along with the PSS-1 pulse to count the number of print-scans (Logic 53.32.01).

The printer circuitry uses a compare counter to identify the type-character that is eligible to print. The compare counter is a six-position, alphameric counter. Its six positions represent the six data bits of a character. The counter advances so that its contents always define the type-character in position to print. The following chart shows the activity of the compare counter during a print-scan.

<u>Print Time</u>	<u>Scan Time</u>	<u>Previous Character</u>	<u>Counter Activity</u>	<u>Resultant Character</u>
Print-Scan 1	PSS-1 mem cycle 1 (home pos)		Reset to 1	1
	PSS-1 mem cycle 2 ↓	1 ↓	Advance by 2 ↓	3 ↓
	PSS-1 mem cycle 44	A	Advance by 2	C
	Between PSS-1 & PSS-2	C	Advance by 2 5 times	1
	PSS-2, mem cycle 1	1	Advance by 1	2
	PSS-2 mem cycle 2 ↓	2 ↓	Advance by 2 ↓	4 ↓
	PSS-2 mem cycle 44	B	Advance by 2	D
	Between PSS-2 & PSS-3	D	Advance by 2 5 times	2
	PSS-3 mem cycle 1	2	Advance by 1	3
	PSS-3 mem cycle 2	3	Advance by 2	5
	PSS-3 mem cycle 44	C	Advance by 2	E
	Between PSS-3 & PSS-1	E	Advance by 2 4 times	1
Print-Scan 2	PSS-1 mem cycle 1	1	Advance by 1	2

When not printing, the compare counter advances by one with each PSS-1 pulse. This provides constant identification of the character at print position 1. A print line can begin with any character at print position 1. The home-reset pulse resets the compare counter to 1 to synchronize the type chain and the compare circuitry.

A subscan is approximately 555 microseconds long. Scanning 44 print positions requires only 484 microseconds (44 memory cycles). A delay latch is set at the end of 44 memory cycles and is reset with the start of the next subscan. The delay time is used to prepare the compare counter for the next subscan by bringing up an advance-by-two gate. This gate is up for one memory cycle (Logic 53.33.05) to allow the required number of advance-by-two pulses to reach the compare counter. The counter-advance pulses are developed on Logic 53.33.06.



\* Prevents printing until chain attains correct speed.  
 ‡ Prevents printing when not scanning out of the buffer.

Figure 4.1-7 Hammer Driver Resets

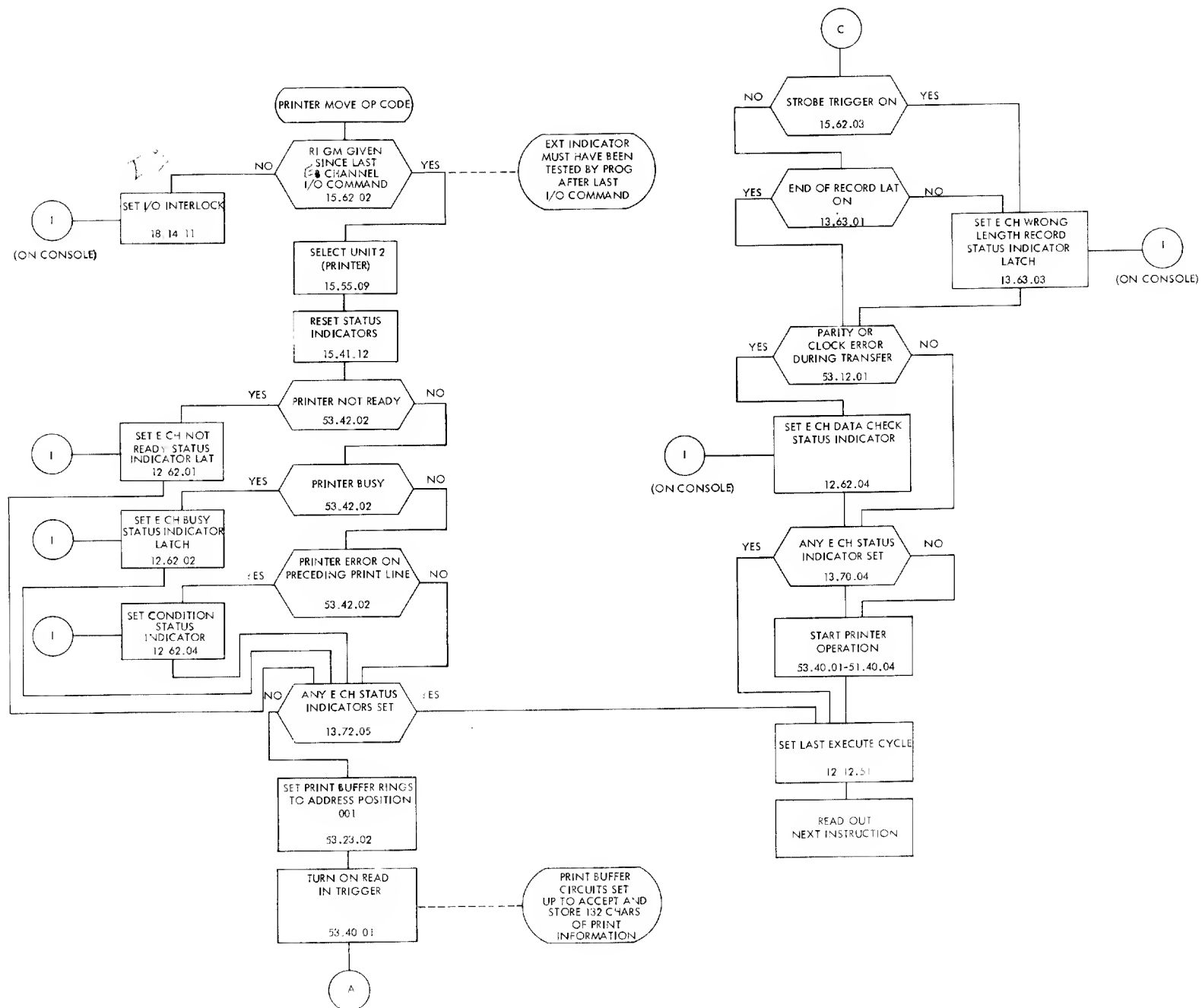


Figure 4.1-8A Printer Move Op Code

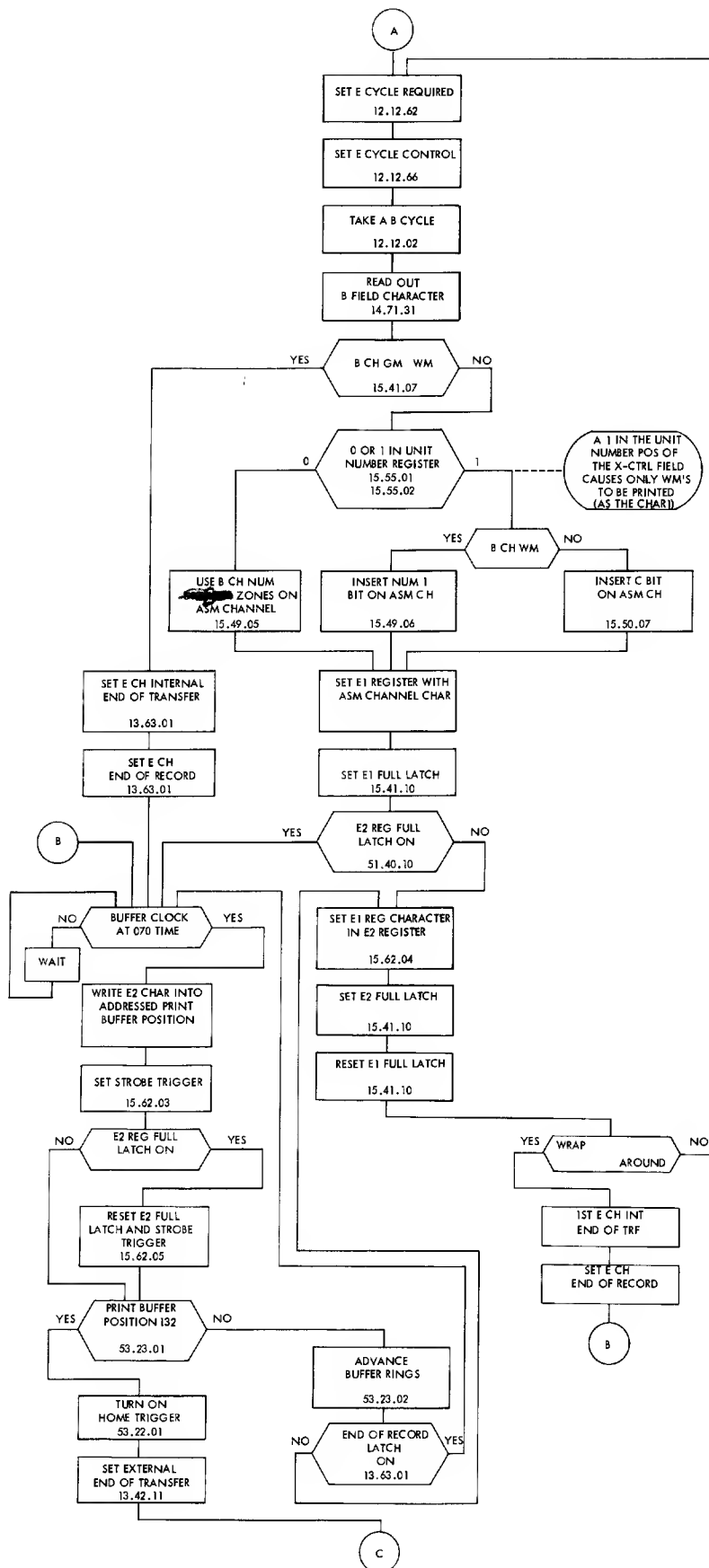


Figure 4.1-8 B Printer Move Op Code

#### 4.1.03 Print-Equal-Compare and Hammer-Fire

To print the correct character in any position, the printer circuitry must be able to detect when the type-character that is aligned to a print position is the same as the character from the corresponding print storage position. A compare matrix (logic 53.34.01-02) compares the bits from the print-storage data register with the bits from the compare counter. If the bit structure of the two characters is the same, the hammer driver for that position is impulsed, and the character is printed. The equal-compare and the hammer-fire row-bit cores (for the addressed position) are set to provide information for error detection on the following scan.

The same three-ring-addressing scheme that selects storage positions is used to select hammer drivers in the driver matrix. A driver can be selected but it is not fired unless an equal-compare condition exists. The chain is moving when printing occurs. Therefore, all hammers should energize for the same length of time for proper print alignment. A latch-type hammer driver produces a timed hammer-magnet impulse for approximately 1.5 milliseconds. The driver-set impulse that an equal-compare condition produces initiates the hammer-magnet current. This current continues until the reset pulse arrives. Ideally, each hammer driver would have its own reset pulse timed to occur exactly 1.5 milliseconds after the set pulse. However, this accuracy is not necessary and six timed-reset pulses reset all of the hammer drivers. The resets provide nominal 1.5 millisecond hammer-magnet impulses. Figure 4.1-7 shows the development of the six reset pulses.

#### 4.1.04 Printer Move-Op Code

Op Code Function (O XXX BBBBB d)

The printer-move Op code transfers a 132-character record from the CPU storage unit to print storage. If the transfer is correct the 1403 printer prints the information in print storage.

Operation (Figure 4.1-8)

Instruction read-out causes the instruction word characters to be placed in various registers in order to control execution of the printer-move Op code.

At I-Op time, the M reads out and is placed into the Op register. The M disregards word marks over characters in storage. This instruction is not given in the load mode, because the 1403 cannot print word-separator characters.

The I-ring advances to I3 and the hundreds position of the X-control field (channel-select character) reads out. The I-ring advances to I3 and the character is placed in the I/O channel-select register. Because the printer is assigned to the E-channel, this character is either a % for unoverlap or a @ for overlap. At I3 time, the CPU tests the E-channel move and load latches. One of these latches was set ON with the last E-channel I/O instruction, and should have been turned OFF with an RId command. If either of these latches is ON, it indicates that the program failed to test the E-channel status-indicators since the last E-channel I/O instruction; and the I/O interlock sets to stop the CPU. If there is no I/O interlock, instruction read-out continues.



At I4, the E-channel status-indicator latches and the E-channel control latches reset in preparation for the execution of the print Op code. Unit-select character 2 reads out and is placed into the E-channel unit-select register. The 2 in the unit-select register addresses the printer, and gates the printer status to the CPU in preparation for setting the status indicators later in I-phase.

At I5 time, the unit-number character reads out and is placed in the E-channel unit-number register. If the character is a 0, word marks are disregarded, and only characters transfer. If the unit-number character is a 1, assembly controls are set up to convert WM's to numerical 1's. Every other character is changed to a C-bit, which prints as a blank.

At I6 through I10, the address of the high-order position of the record to be punched reads out and is placed in the B- and D-address registers for an unoverlapped command, or in the B-, D- and E-address registers for an overlapped command. The B-address register scans the storage-output field on an unoverlapped transfer, while the E-address register scans the storage-output field.

The Op modifier character reads out at I11 time and is set into the Op modifier register. For the printer Op code, the Op modifier character is a W to specify a write, or output, operation.

The CPU normally takes another I-cycle to search for the word mark in the next position. During this last instruction read-out cycle, a status sample A-pulse develops to set the appropriate status indicators with the status of the printer. Printer Not Ready, Printer Busy or Error-on-Previous-Print-Line set respectively the E-channel-not-ready, E-channel-busy, and E-channel-condition status indicators. If any of these indicators are set at this time, the CPU does not execute the instruction. Instead it reads out the next instruction in sequence. The print-error circuit is reset following the status sample A-pulse, because the error is now stored in the E-channel-condition status indicator. If no indicators are set at status sample A-time, it indicates that the printer is able to execute the command.

If the command is overlapped, the Op register, Op mod register, and the I/O channel-select register must be free to accept the characters of the next instruction word. The information in these registers is translated and stored in the E-channel move or load, and input or output latches to control the execution of this I/O command, while the registers are free for the following instruction. This takes place at I12 or last-instruction-read-out cycle.

A delayed status sample A-pulse sets either the overlap or unoverlap-in-process latch (providing no indicators are ON) and the print-storage circuitry is conditioned to accept the record to be printed.

E-cycles are required to transfer characters from core storage to the E-data registers. The CPU takes E-cycles as required to keep both E1 and E2 registers full. The print-storage clock runs continuously to write the E2 characters into the addressed print-storage positions. At the end of each print-storage memory cycle, the CPU is notified that another character is required in the E2 register. The character in E1 transfers to E2 and an E-cycle is taken to refill E1.

The print-storage memory cycle is 11 microseconds. E-cycles on this Op code are 5.25 microseconds long. During the time when both E-registers are full, no E-cycle is required and the CPU clock stops. When the group-mark with a word-mark is sensed, it is not set into E1. Further E-cycles are prevented and the end-of-record condition occurs. The character remaining in E2 when the end-of-record condition occurs is the last or 132nd character. It is written into the last or 132nd position of print storage to define the external end-of-transfer. If there is a wrong-length-record condition, the E-channel wrong-length-record status indicator is set. An error in the printer circuitry (parity or clock error) sets the E-channel data-check status indicator. Either indicator ON causes the CPU to read out the next instruction. Failure to print the line is detected when the RId command occurs later in the program.

If there are no indicators set, the printer starts and the next instruction reads out.

A 100-print-position 1403 requires special circuitry to prevent parity errors and to make the wrong-length-record check effective. The entire print storage (132 positions) is read into, and scanned out of, regardless of the number of print positions in the 1403. However, the transfer from the CPU to print storage of a 100 print-position machine must be 100 characters in length, and a group mark with a word mark must be in the 101st storage position. To prevent parity errors at the extra print-storage positions (101-132), a C-bit insert-circuit detects when print-storage position 100 is addressed, and it inserts C-bits in all subsequent positions.

For a 100-position printer, the circuitry that detects wrong-length-records is signalled when print storage has reached position 100, because each record transferred from the CPU must be 100 characters in length. If the group-mark with a word-mark is in any other than the position to the right of the 100-character record in core storage, the E-channel wrong-length status indicator is set.

## Circuits

Op-code grouping lines are:

Percent type op codes  
 Not addr dbl op codes  
 2 Addr plus mod op codes  
 2 Address op codes  
 Addr type op codes  
 No index on 1st addr ops  
 No-branch op codes  
 M or L op codes

1. Gate the printer status to CPU.

<u>Signal</u>	<u>Control</u>	<u>Logic</u>
E-ch select any buffer	E-ch select unit 2	13. 60. 03
(not) Buffer ready	Forms not-ready status. Print not-ready status	51. 40. 18

<u>Signal</u>	<u>Control</u>	<u>Logic</u>
(not) E-ch ready bus	Buffer ready E-ch select any buffer	12. 62. 01
Busy	Forms busy status Print busy status	51. 40. 19
E-ch busy bus	Buffer busy E-ch select any buffer	12. 62. 02
Buffer conditions	Print-error status	51. 40. 20
E-ch condition bus	Buffer conditions E-ch sel any buffer	12. 62. 03

2. Set the E-channel status indicators.

<u>Signal</u>	<u>Control</u>	<u>Logic</u>
E-ch status sample A	M or L op codes I/O percent latch 112, LGD	13. 65. 06
E-ch not ready	(not) E-ch ready bus E-ch status sample A	12. 62. 01
E-ch busy	E-ch busy bus E-ch status sample A	12. 62. 02
E-ch condition	E-ch condition bus E-ch status sample A	12. 62. 04

3. If any indicator is set at status sample A-time, go on to the next instruction in sequence.

<u>Signal</u>	<u>Control</u>	<u>Logic</u>
E-ch correct-length record	Set E-ch condition or Set E-ch busy or Set E-ch not ready	13. 63. 03
E-ch any status (on)	E-ch correct-length record	13. 72. 05
Last-execute-cycle I/O	E-ch any status (on) I/O percent latch Last insn RO cycle (not) File op	13. 65. 07
Last-execute cycle	Last-execute cycle I/O	12. 12. 51

4. If the condition latch is ON, reset the print-error latch.

<u>Signal</u>	<u>Control</u>	<u>Logic</u>
E-ch status sample A-delay	Logic gate E B-ch WM	13. 65. 06

<u>Signal</u>	<u>Control</u>	<u>Logic</u>
	M or L op codes I-ring 12 time (not) E-ch select unit F Percent or com1 at I-cycle	
Reset select buffer latches	E-ch condition E-ch select any buffer E-ch status sample A-delay	13. 70. 04
Reset CPU	Reset select buffer latches	51. 40. 06
Printer error (off)	Select unit 2 powered Reset CPU	53. 12. 01

5. If no status indicators are ON, request an E-cycle.

<u>Signal</u>	<u>Control</u>	<u>Logic</u>
E-cycle required	E-ch output mode (not) E-ch int end-of-transfer (not) E1 reg full (not) E-cycle any last gate E-ch-in-process	12. 12. 62
E-cycle control	E-cycle required 2nd clock pulse Logic gate Z (not) E-ch int end-of-transfer (not) F-cycle required	12. 12. 66

6. Start the appropriate logic clock and transfer the contents of the proper address register to STAR. If the instruction is unoverlapped, start the normal logic clock (A-K) and read out BAR.

<u>Signal</u>	<u>Control</u>	<u>Logic</u>
(start logic clock)	E-cycle required E-ch unovlp-in-process	11. 10. 10
RO BAR	E-ch unovlp-in-process E-cycle ctrl (not) Console inhibit AR RO (not) 1st I/O cycle control (not) E-ch 2nd addr unovlp	14. 71. 31

7. Start the appropriate logic clock and transfer the contents of the proper address register to STAR. If the instruction is overlapped, start the logic clock extension ring (R-Y) and read out EAR.

<u>Signal</u>	<u>Control</u>	<u>Logic</u>
(start logic extension clock)	E-cycle required E-ch overlap-in-process	11. 10. 20

<u>Signal</u>	<u>Control</u>	<u>Logic</u>
RO EAR	E-cycle ctrl E-ch ovlp-in-process (not) Console inhibit AR RO	14. 71. 35

8. Take an E-cycle to read out the addressed character.

<u>Signal</u>	<u>Control</u>	<u>Logic</u>
E-cycle	E-cycle ctrl Logic gate B or S	12. 12. 66

9. If the unit number in the instruction word is 0, gate the B-channel character through assembly.

<u>Signal</u>	<u>Control</u>	<u>Logic</u>
Output-field cycle	E-ch cycle E-ch select unit 2 E-ch unit nu 0	15. 49. 04
Use B-ch nu	Output cycle Output-field cycle	15. 49. 05
E-ch select odd-parity unit	E-ch sel unit 2	13. 60. 03
Odd parity cycle	B-cycle I/O percent of lozenge E-ch select odd-parity unit	13. 60. 02
Use B-ch zones	Output cycle Output-field cycle Odd-parity cycle	15. 49. 05
Assembly ch__bit	Use B-ch num Use B-ch zones B-data req__bit	15. 50. 01-10

10. If the unit number in the instruction word is 1, insert 1 bits for WM's and C-bits for everything else.

<u>Signal</u>	<u>Control</u>	<u>Logic</u>
Output WM cycle	E-ch cycle E-ch select unit 2 E-ch unit number 1	15. 49. 02
Assembly ch nu one insert	Output WM cycle Output cycle B-ch WM bit	15. 49. 06
Use no numerics	Output cycle Output WM cycle B-ch not WM bit	15. 49. 01
Use no zones	Output cycle Output WM cycle	15. 49. 01
Use no numerics and use no zones	Combine with B-ch not WM bit to produce a C-bit	

11. Gate the assembly-channel character to E1 input.

<u>Signal</u>	<u>Control</u>	<u>Logic</u>
Gate asm ch to E1 input	E-ch output mode (not) Control reg disable (not) 1401 B-op code	15. 62. 07

12. If B-channel character is a group-mark with a word-mark, end the internal transfer.

<u>Signal</u>	<u>Control</u>	<u>Logic</u>
B-ch group mark • WM	B-ch 1• 2• 4• 8• A• B• WM	15. 41. 07
E-ch end-of-record latch	E-cycle B-ch GM • WM Logic gate F W-symbol op modifier (not) 1401 cd-print-in-process or E-cycle B-ch GM•WM Logic gate W	13. 63. 01
E-ch int end-of-transfer	(same as E-ch end-of-record)	13. 63. 01

13. Set the E1 data register.

<u>Signal</u>	<u>Control</u>	<u>Logic</u>
Set E1	E-cycle E-ch output mode (not) E-ch 1st char 2nd addr B-ch not group mark • WM Logic gate F or W	15. 62. 04

14. Transfer E1 to E2.

<u>Signal</u>	<u>Control</u>	<u>Logic</u>
E-ch reset	I-ring 4 time Percent or coml at Logic gate C	15. 41. 12
E1 reg not word separator	E-ch reset	15. 41. 11
Copy E1 BCD to E2	E1 reg not word separator	15. 62. 06
E2 reg full latch (off)	E-ch reset	15. 41. 10
Set E2 reg	E1 reg full latch (on) E2 reg full latch (off)	15. 62. 04
E2 reg full latch (on)	E-ch output mode Set E2 reg	15. 41. 10
E1 reg full latch (off)	E1 reg not word separator Set E2 reg	15. 41. 10

When the E1 reg full latch is turned off, the E-cycle required is brought up to cause another character to read out and be gated to the E1 register.

15. Request a print-transfer scan.

<u>Signal</u>	<u>Control</u>	<u>Logic</u>
Ready to buffer	E-ch-in-process E-ch select any buffer E2 reg full	13. 70. 04
Ready	Ready to buffer	51. 40. 04
Read in	Ready Ready for CPU transfer Select unit 2	53. 40. 01
Print trans	Read in (not) CE mode (not) C-bit insert	53. 11. 05
Trans scan	Print trans	51. 40. 40

16. Write the E2 character into the addressed print-storage position.

<u>Signal</u>	<u>Control</u>	<u>Logic</u>
CPU to I/O sync___bit	E2 reg___bit	15. 60. 31-34
CPU to I/O bit___	CPU to I/O sync___bit	51. 40. 10
Inh gate___bit	CPU to I/O bit___	51. 16. 03
Write pulse	070-100	51. 30. 05

17. Set the strobe trigger.

<u>Signal</u>	<u>Control</u>	<u>Logic</u>
Strobe latch	Trans scan Time 020-030	51. 40. 43
CPU to I/O trans	Print trans	51. 40. 40
Buffer strobe	Strobe latch CPU to I/O trans Time 100-000	51. 40. 43
E-ch strobe trigger	Buffer strobe E-ch select any buffer	15. 62. 03
Reset E2 full latch	E2 reg full E-ch strobe trigger E-ch output mode	15. 62. 65
E2 reg full latch (off)	Reset E2 full latch	15. 41. 10
E-ch strobe trigger (off)	Reset E2 full latch E-ch output mode	15. 62. 03

18. After 132 memory cycles, stop scanning print-storage.

<u>Signal</u>	<u>Control</u>	<u>Logic</u>
Turn off rings latch (on)	Read in	53. 23. 01
Gate on home trigger	Turn off rings latch (on) Threes ring 3 Fives ring 5 Tens ring 3	53. 23. 01
Read in (off)	Gate on home trigger Time 000-010	53. 40. 01
End-of-transfer	Read in (off) (not CE mode)	53. 40. 01
Home trigger	End-of-transfer Gate on home trigger	53. 22. 01

19. Set the external end-of-transfer.

<u>Signal</u>	<u>Control</u>	<u>Logic</u>
I/O op	Select unit 2	51. 40. 01
End-of-transfer 1 latch	I/O op Trans scan	51. 40. 12
(not) Print trans	Read in (off)	53. 11. 05
(not) Trans scan	(not) Print trans	51. 40. 40
End-of-transfer 2 latch	End-of-transfer 1 (not) Trans scan Time 090-100	51. 40. 12
Buffer end-of-transfer	End-of-transfer 2	51. 40. 12
E-ch ext end-of-transfer	Buffer end-of-transfer E-ch select any buffer	13. 42. 11

20. Test for a wrong-length record and set the appropriate E-channel status indicator.

<u>Signal</u>	<u>Control</u>	<u>Logic</u>
E-ch int end-of-trf delayed	E-ch int end-of-transfer Logic gate Z	13. 65. 05
E-ch status sample B	E-ch int end-of-trf delayed Logic gate Z E-ch ext end-of-transfer (not) 2nd clock pulse	13. 65. 05
E-ch wrong-length record	E-ch status sample B E2 reg full or	13. 63. 03



<u>Signal</u>	<u>Control</u>	<u>Logic</u>
	(not) E-ch end-of-record latch or E-ch strobe trigger	
E-ch correct-length record	E-ch status sample B E-ch end-of-record latch (not) E1 reg full (not) E2 reg full (not) E-ch strobe trigger	13. 63. 03

21. If a print-storage error occurs during transfer, set the E-channel check indicator.

<u>Signal</u>	<u>Control</u>	<u>Logic</u>
Printer error	C-bit check Pwr clock error	53. 12. 01
Print trans error status	Printer error Select unit 2	53. 42. 02
Buffer error	Print trans error status	51. 40. 21
E-ch check bus	Buffer error E-ch select any buffer	12. 62. 03
E-ch check	E-ch check bus E-ch status sample B	12. 62. 04

22. If the instruction is unoverlapped, set the last execute cycle.

<u>Signal</u>	<u>Control</u>	<u>Logic</u>
Last execute cycle I/O	E-ch unovlp-in-process (not) 1401 card or print op code E-ch status sample B	13. 65. 07
Last execute cycle	Last execute cycle I/O	12. 12. 51

23. If no indicators are set, start the print operation.

<u>Signal</u>	<u>Control</u>	<u>Logic</u>
E-ch second sample B	E-ch status sample B (not) 2nd clock pulse	13. 65. 05
Correct trans to buffer	E-ch correct-length record (not) E-ch check E-ch select any buffer E-ch second sample B	13. 70. 04
Go	Correct trans to buffer	51. 40. 04
Scan cell	Go Select unit 2 powered Ready for print or forms	53. 40. 01

24. Reset the print-transfer error (if ON).

<u>Signal</u>	<u>Control</u>	<u>Logic</u>
Reset select buffer latches	E-ch sel any buffer F-ch second sample B	13.70.04
Reset CPU	Reset select buffer latches	51.40.06
Printer error (off)	Reset CPU Select unit 2 powered	53.12.01

25. Reset the overlap or unoverlap-in-process latch.

<u>Signal</u>	<u>Control</u>	<u>Logic</u>
E-ch status sample B-delay	E-ch second sample B (not) 2nd clock pulse	13.65.05
E-ch overlap-in-process (off)	E-ch status sample B-delay	13.60.04
E-ch unoverlap-in-process (off)	E-ch status sample B-delay	13.60.04

## 4.2.00 CARRIAGE CONTROL

### 4.2.01 Carriage Circuits

#### Carriage-Control Character Register and Decode

A forms control instruction is included in stored programs to provide program control of carriage spacing and skipping. This instruction sends a single carriage-control character (CCC) to the printer circuitry, where the character is gated to the CCC register. A decode circuit reconstructs the character from the bits present in the CCC register to determine the operation that is to be performed by the carriage.

The CCC register consists of six latches to store the six data bits, B, A, 8, 4, 2, 1 (logic 53.50.01). The six bits are gated by a CCC register gate that signifies that the carriage is ready to accept a character, and that a forms-control instruction is called for.

#### Carriage Fast and Stop Controls

The carriage is a two-speed device. Speed depends on the operation that is being performed. All space operations take place at slow speed. The speed of all skip operations (including manual restore) is governed by the length of the skip. Skips of eight spaces or greater start at high speed. The last seven spaces of any skip are made at slow speed. Thus, if a skip is seven spaces or less, it is made at slow speed. The condition of the channel latches control skip speeds. Each latch is set by the stop brush and reset by the slow brush that corresponds to that position (Figure 4.2-1).

#### Carriage Magnet Control (Figure 4.2-2)

There are four carriage magnets, labelled Space On and Space Off, and Skip On

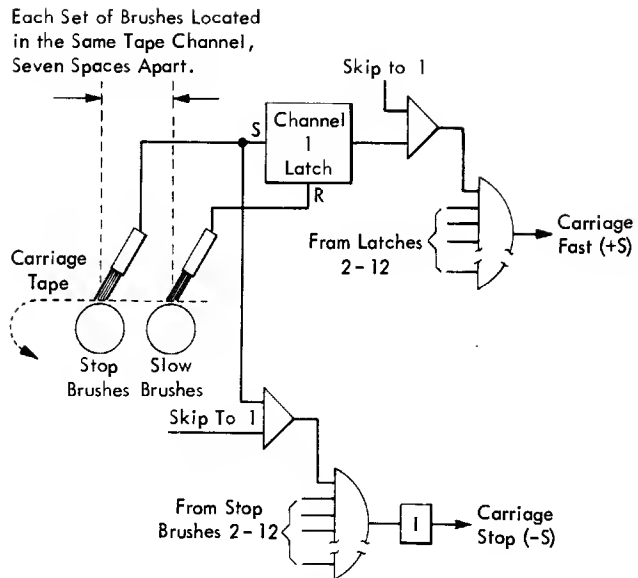


Figure 4.2-1 Carriage Fast and Stop Controls

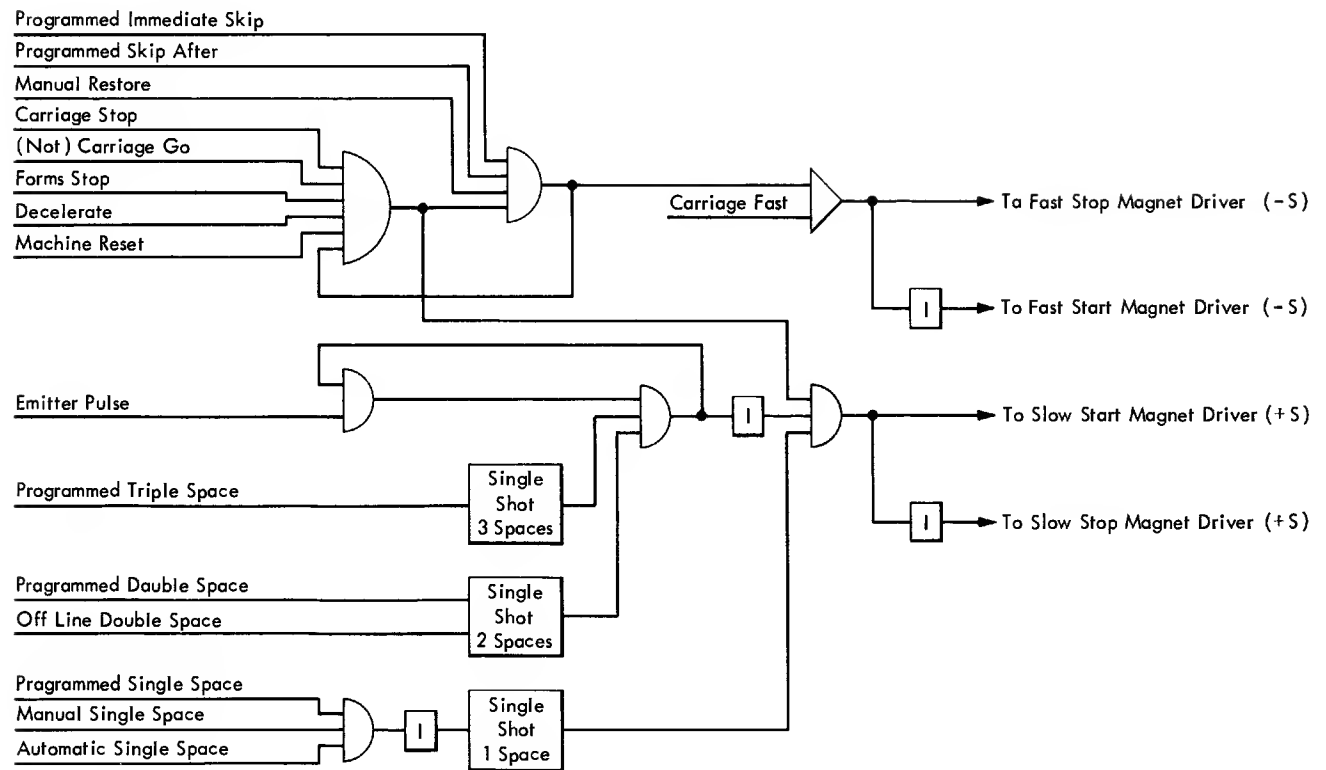


Figure 4.2-2 Carriage Magnet Control

and Skip Off. For a slow speed operation, the space-on and skip-off magnets energize. During a high-speed operation, both on magnets must energize.

The space-on magnet energizes whenever the carriage is set in motion. The skip-on magnet energizes only for skips of eight or more spaces.

Spacing is accomplished by holding the space-on magnet energized until the form moves the required distance (1, 2 or 3 spaces). The space-on magnet receives an impulse the duration of which depends on the number of spaces desired. The impulse is formed by firing one of three single shots (Figure 4.2-2).

With a skip of eight spaces or more, the carriage-fast line is up and the carriage stop line is down (Figure 4.2-1). This energizes both space-on and skip-on magnets. When the slow brush signals that seven spaces remain in the skip, the carriage-fast line goes down. This removes the drive to the skip-on magnet, and impulses the skip-off magnet. However, the carriage stop line is still down; this maintains the space-on magnet-drive. The skip continues at slow speed until the brush senses the addressed tape-channel hole and brings up the carriage stop line.

With a manual-restore operation, the addressed tape channel is 1, and the carriage skips until the hole in channel 1 is sensed. A restore operation can take place at either high or slow speed, depending on the distance to be skipped.

Channel Indicators 9 and 12 (Logic 53.56.01)

Two channels in the carriage tape determine the forms position. Impulses from the stop brushes for channels 9 and 12 set latches that the stored program can interrogate. Sensing holes in tape channels 9 and 12 sets corresponding channel-indicator latches. The next hole that is read in any tape channel resets the latches.

A forms stop interlocks the 1403, and manual intervention is required to restore the printer to the ready status.

#### 4.2.02 Forms Control Op Code

##### Op-Code Function (Fd)

A single character is sent to the carriage circuitry to control forms skipping and spacing. The bit structure of this character signals the tape-controlled carriage either to take a single, double or triple space, or to skip to the next hole in a designated tape channel. The spacing and skipping can be executed either immediately or after the next print command.

##### Operation (Figure 4.2-3)

Decoding an F-Op code selects the E-channel to send a single carriage-control character (the Op modifier) to the printer circuitry. An I/O interlock check stops the CPU, if the E-channel status indicators have not been tested by an RI # command since the last E-channel I/O command. The I/O interlock check develops as follows. Every E-channel I/O command turns on either the move or load latch for that channel. The latch remains on until it is turned off by an RI # command. The fol-

248

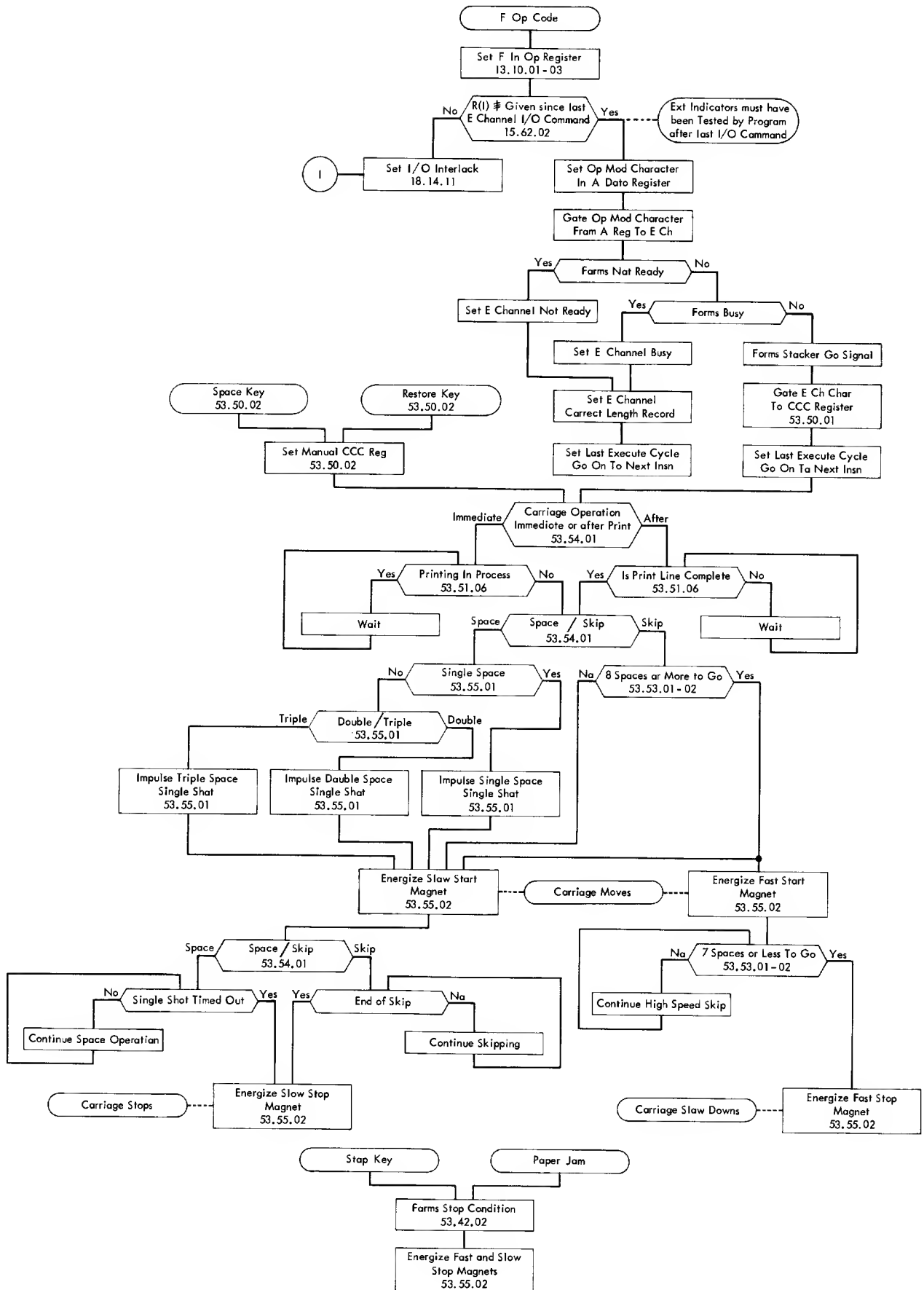


Figure 4.2-3 Forms Op Code

lowing E-channel I/O command (in this case, Fd) checks both the move and load latches for an OFF condition. If either latch is ON at I-ring Op time, an I/O interlock check results.

After the I/O interlock check, the E-channel data register and status-indicator latches reset in preparation for sending the Op-modifier character to the printer circuitry. This character reads out of storage and is placed in the A-data register. From the A-data register, the character is gated through the assembly area to the E-data registers. A Select Unit 2 signal gates the printer and carriage conditions that affect the forms Op code to the E-channel status indicators. A printer not-ready condition (chain not running, out of forms, etc.) sets the E-channel not-ready latch. If a previous carriage operation is not complete, the E-channel busy latch is set.

These status indicators are not tested to determine whether the carriage and printer are able to execute a forms command. If either the E-channel not-ready or the E-channel busy latch is ON, the forms command is bypassed. Failure to execute this command is detected when an RId command tests the indicators. Both latches OFF allows a forms-stacker go signal to start the carriage operation. The modifier character is gated to the carriage-control character register where it is decoded to determine the type of carriage operation that is to be performed.

If printing is taking place when the forms command is initiated, the carriage operation is delayed until the end of the print line. In addition, if the forms command is an after-print command, the carriage operation is delayed until the end of the next print line.

The carriage control character register resets when the forms operation completes in order to prevent the operation from repeating.

## Circuits

Op-code grouping lines are:

Not addr dbl op codes  
2 char only op codes  
No branch op-codes  
Op mod to A-ch or B-cycle ops  
Regen mem or B-cycle op codes

### I. Set up controls for forms Op.

<u>Signal</u>	<u>Control</u>	<u>Logic</u>
Set op register	I-op	12. 13. 04
Forms ctrl to buffer	Forms ctrl Op code	13. 70. 04

### 2. Test for I/O interlock.

<u>Signal</u>	<u>Control</u>	<u>Logic</u>
(not) E-ch interlock	E-ch move/load Latches not set	15. 62. 02
(not) I/O interlock check	(not) E-ch interlock	18. 14. 11

3. Reset the E-channel controls and status latches.

<u>Signal</u>	<u>Control</u>	<u>Logic</u>
F or K E-ch reset	Forms ctrl op code	15. 41. 12
E-ch reset	F or K E-ch reset	15. 41. 12

4. Select unit 2 (printer). The F-Op code forces a select-unit-2 signal to gate status-information to the CPU.

<u>Signal</u>	<u>Control</u>	<u>Logic</u>
E-ch U-sel-reg-2 bit	F-ctrl op code	15. 55. 09
E-ch sel unit 2	E-ch U-sel-reg-2 bit	13. 50. 03

5. Set the Op-modifier character in A-data register.

<u>Signal</u>	<u>Control</u>	<u>Logic</u>
Sw B-ch to A-reg	1-cycle	15. 38. 01

6. Gate the A-channel character (Op modifier) to E2 register.

<u>Signal</u>	<u>Control</u>	<u>Logic</u>
E-ch move mode	Last insn RO cycle 2 Char only op codes	15. 62. 02
A-ch (any) bit	Sw B-ch to A-reg	15. 39. 01-08
Use A-ch num	I-cycle	15. 49. 03
Use A-ch zone	I-cycle	15. 40. 03
Asm ch (any) bit	Use A-ch num Use A-ch zones	15. 50. 01-10
Gate asm ch to E1 input	I-cycle	15. 62. 07
Set E1	2 Char only op codes	15. 62. 06
E1 not WS	Set E1 Not WM or WS	15. 41. 12
Copy E1 BCD to E2	E1 not WS	15. 62. 06
Set E2 reg	E1 full E2 not full	15. 62. 04
E2 reg___ bit	Set E2 reg	15. 60. 31-34

7. Gate the forms status to the CPU.

<u>Signal</u>	<u>Control</u>	<u>Logic</u>
(not) Forms busy	(carriage not in use)	53. 50. 03
(not) Buffer busy	(not) Forms busy	51. 40. 19
Ready for print or forms	(printer and carriage ready)	53. 42. 02
(not) Print not-ready status	Select unit 2 Ready for print or forms	53. 42. 02
Buffer ready	(not) Forms not-ready status	51. 40. 18

8. Set the E-channel status-indicators according to the forms status.

<u>Signal</u>	<u>Control</u>	<u>Logic</u>
E-ch status sample A	2 Char only op codes	13. 65. 06
E-ch not ready	(not) Buffer ready E-ch status sample A	12. 62. 01
E-ch busy	Buffer busy E-ch status sample A	12. 62. 02

9. Test the status indicators and develop the forms stacker go signal.

<u>Signal</u>	<u>Control</u>	<u>Logic</u>
Last insn RO cycle	2 Char only op codes I-ring 2 B-ch WM bit	12. 13. 05
Forms stacker go	(not) E-ch not ready (not) E-ch busy (not) E-ch no transfer 2 Char only op codes Last insn RO cycle Logic gate E	13. 70. 04

10. Gate the E2 register character (op modifier) to the carriage-control character register.

<u>Signal</u>	<u>Control</u>	<u>Logic</u>
CCC reg gate	Forms and stacker go Forms control Reads for print or forms (not) Forms busy status to CPU	53. 50. 03
CPU to I/O___bit	E2 reg___bit	15. 60. 31-34
CCC reg___bit	CCC reg gate CPU to I/O___bit	53. 50. 01-02

11. Develop the magnet impulses for spacing.



<u>Signal</u>	<u>Control</u>	<u>Logic</u>
Start forms op	PS - 48	53. 32. 02
Print-in-process latch (on)	Mach reset	53. 51. 06
Delay forms latch (on)	Mach reset	53. 51. 06
Carr go	Print-in-process latch (on) Delay forms latch (on) CCC reg char	53. 51. 06
CCC decode 1, 2, or 3	CCC reg___bit	53. 51. 03
Space single shot	Space gate CCC decode 1, 2, or 3	53. 55. 01
Slow space control	Space single shot	53. 55. 01
Space magnet ON to 1403	Slow space control	53. 55. 02
(not) Space magnet OFF to 1403	Slow space control	53. 55. 02

12. Develop the magnet impulses for skipping.

<u>Signal</u>	<u>Control</u>	<u>Logic</u>
CCC decode___	CCC reg___bit	53. 51. 03
Channel___latch (off)	(8 or more spaces to go)	53. 53. 01-02
Carr fast	CCC decode___ Channel___latch (off)	53. 51. 08
Fast skip control	Carr fast Carr go CCC reg ( $\bar{B} \cdot \bar{A}$ or $B \cdot A$ ) bits	53. 54. 01
Slow skip control	Carr fast Carr go	53. 54. 01
Fast magnet ON to 1403	Fast skip control	53. 55. 02
(not) Fast magnet OFF to 1403	Fast skip control	53. 55. 02
Space magnet ON to 1403	Slow space control	53. 55. 02
(not) Space magnet OFF to 1403	Slow space control	53. 55. 02
Channel___latch (on)	(Less than 8 spaces remaining)	53. 53. 02
Carr fast (off)	Channel___latch (on)	53. 51. 08
Fast skip control (off)	Carr fast (off)	53. 54. 01
(not) Fast magnet ON to 1403	Fast skip control (off)	53. 54. 01

<u>Signal</u>	<u>Control</u>	<u>Logic</u>
Skip magnet OFF to 1403	Fast skip control (off)	53.55.02
CCC decode stop A	(Skip completed)	53.51.05
Carr stop	CCC decode stop A	53.51.04
Slow skip control (off)	Carr stop	53.54.01
Slow space control (off)	Slow skip control (off)	53.55.01
(not) Space magnet ON to 1403	Slow space control (off)	53.55.02
Space magnet OFF to 1403	Slow space control (off)	53.55.02

### 13. Reset the CCC register.

<u>Signal</u>	<u>Control</u>	<u>Logic</u>
Reset delay latch (on)	Start forms op	53.55.03
Decelerate	Slow space control (off)	53.55.03
CCC reg reset	Reset delay latch (on) Decelerate	53.55.03

## 4.3.00 CHECKING FEATURES

### 4.3.01 Printer Synchronization Check (Logic 53.43.01)

To print correct information, the printer-compare circuitry synchronizes with the type chain. Due to mechanical linkage between the chain and the timing disk, the 1 type-slug always aligns at print position 1 when the home pulse occurs. The compare counter should always contain a 1 and the PSS-1 trigger should be ON at this time, if they are properly synchronized with the chain. If the compare counter contains anything but a 1-bit at home-pulse time, a sync check sets the indicator latch (Figure 4.3-1). In addition, if printing is taking place when the sync check occurs, the print-error latch sets to show that an error occurred.

### 4.3.02 Printer Parity Check (Logic 53.44.01-02)

Information is stored in print storage in odd-parity form. Every character that reads into storage is checked by the printer parity-check circuit. The check is made at the input of the inhibit drivers.

Two parity-check triggers are reset OFF near the beginning of the memory cycle (Figure 4.3-2). During the write portion of each memory cycle, any of the seven bits that are present at the inputs of the inhibit drivers are switched to the trigger inputs. Binary trigger inputs allow flipping ON and OFF with successive set pulses.

The 1, 4, A and C-bits switch to TR-1 at 6, 7, 8, and 9 times, respectively. The 2, 8, and B-bits switch to TR-2 at 6, 7, and 8 times, respectively. If an even

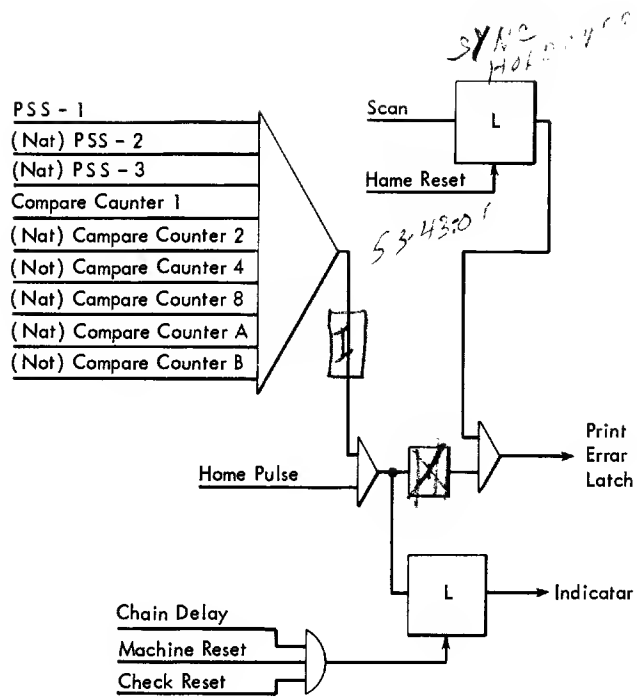
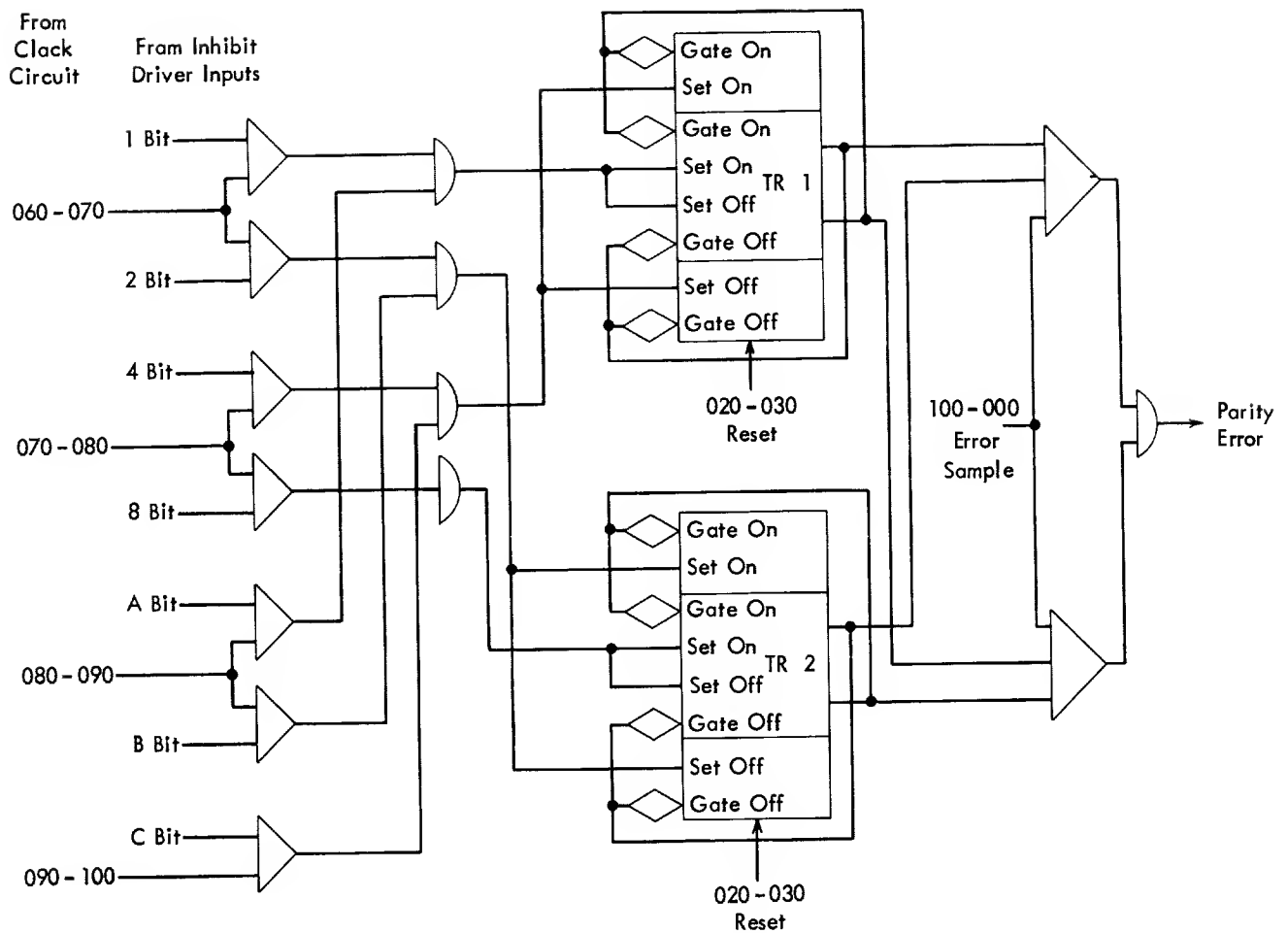


Figure 4.3-1 Printer Synchronizatian Check



A character containing C, B, and 1 bits is checked. No error results since the triggers are opposite at error sample time.

A character containing B and 1 bits is checked. A C-bit check occurs since the triggers are the same at error sample time.

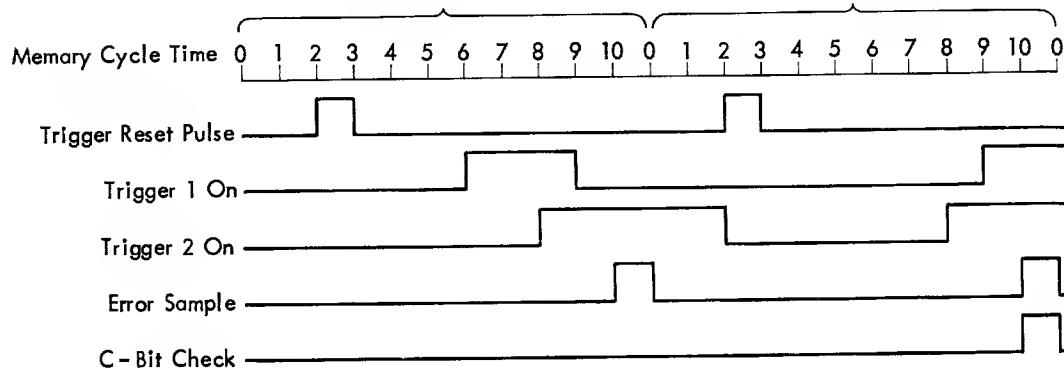


Figure 4.3-2 Printer Parity Check

number of bits is at the inhibit-driver inputs, the triggers are either both ON or both OFF at ten-time when the error sample occurs, and a C-bit check results. An odd number of bits causes the triggers to be opposite (not both ON or OFF) when they are tested at ten-time. Therefore, the C-bit check signal is not generated.

#### 4.3.03 Print-Error Detector

Four core planes of print storage detect and store errors. These are: print-line-complete, equal-compare, hammer-fire row-bits and hammer-check (Figure 4.3-3).

The print-line-complete cores set during a CPU transfer for those print positions that contain a printable character. During a memory cycle, an equal-compare blocks the regeneration of the print-line-complete core, and sets the equal-compare core for that position. If an equal-compare occurs for a position that does not contain a printable character, an error is indicated, and the hammer-check core is set for that position (Figure 4.3-3).

An equal-compare condition impulses the hammer driver that sets the corresponding hammer-fire row-bit core. If an equal-compare core is set, the corresponding hammer-fire row-bit core should be set, and vice versa. If either core sets without the other, a print error results on the following print scan when that position again reads out. The error sets the corresponding hammer-check core.

After the 48th scan, all positions that contained a printable character should be printed, and no print-line-complete core should remain set. Any print-line-complete core that remains ON when the 49th scan occurs causes a print error.

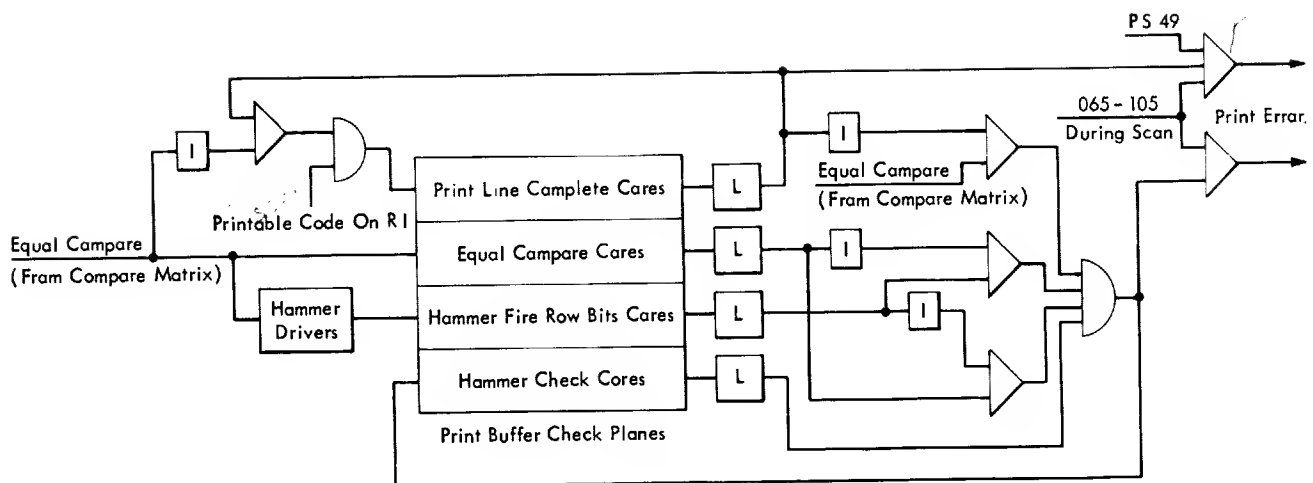


Figure 4.3-3 Print Error Detector

*When Hammer is fired it is a sign of Print Line Complete, and all Hammer fire are*

## 5.0.00 MAGNETIC TAPE CONTROL

### 5.1.00 MAGNETIC-TAPE-OPERATION INSTRUCTIONS

There are three basic types of magnetic-tape operations: reading, writing, and unit control. Reading and writing can be done with or without word marks, depending upon the instruction that is used. The unit control instruction initiates backspacing tape, writing tape marks, rewinding tape reels, and skipping over defective areas on tape. The instruction format is included in the basic operation where it is used.

#### 5.1.01 Tape Write Operation (Figure 5.1-1)

The information from the specified B-address is written by the selected-tape unit with or without word-separator control.

<u>Instruction Format</u>		
<u>Character</u>	<u>Position</u>	<u>Function</u>
M	Op code	I/O Op codes without word marks
or		
L	Op code	I/O Op codes with word marks
%	X-field 1	Use E-channel unoverlap
or		
@	X-field 1	Use E-channel overlap
or		
□	X-field 1	Use F-channel unoverlap
or		
*	X-field 1	Use F-channel overlap
U	X-field 2	Select tape overation— even parity
or		
B	X-field 2	Select tape operation— odd parity
0-9	X-field 3	Select magnetic-tape unit
	B-address	High-order position of the field
W	d-modifier	Write tape
or		
X	d-modifier	Write tape to end-of-storage

During the write operation the characters in the B-field read out to the B-channel, through assembly, and to the E1 or F1 registers. If the E-channel is selected by the I/O channel-select character of the instruction word X-field, data transfers from the E1 register to the E2 register. The output of the E2 register that is corrected for odd or even parity (odd or even parity is determined by unit-select character of the X-field) conditions the input of the R/W register in the TAU. These input lines are sampled during the TAU write-clock cycle and the data transfers into the R/W register. The output of the R/W register conditions the write triggers in the selected-tape unit, and the TAU-generated write-pulse transfers the data onto the tape.

Before the data reads out of storage, control circuits must be established and certain status checks must be performed. An early sample of the TAU-busy status

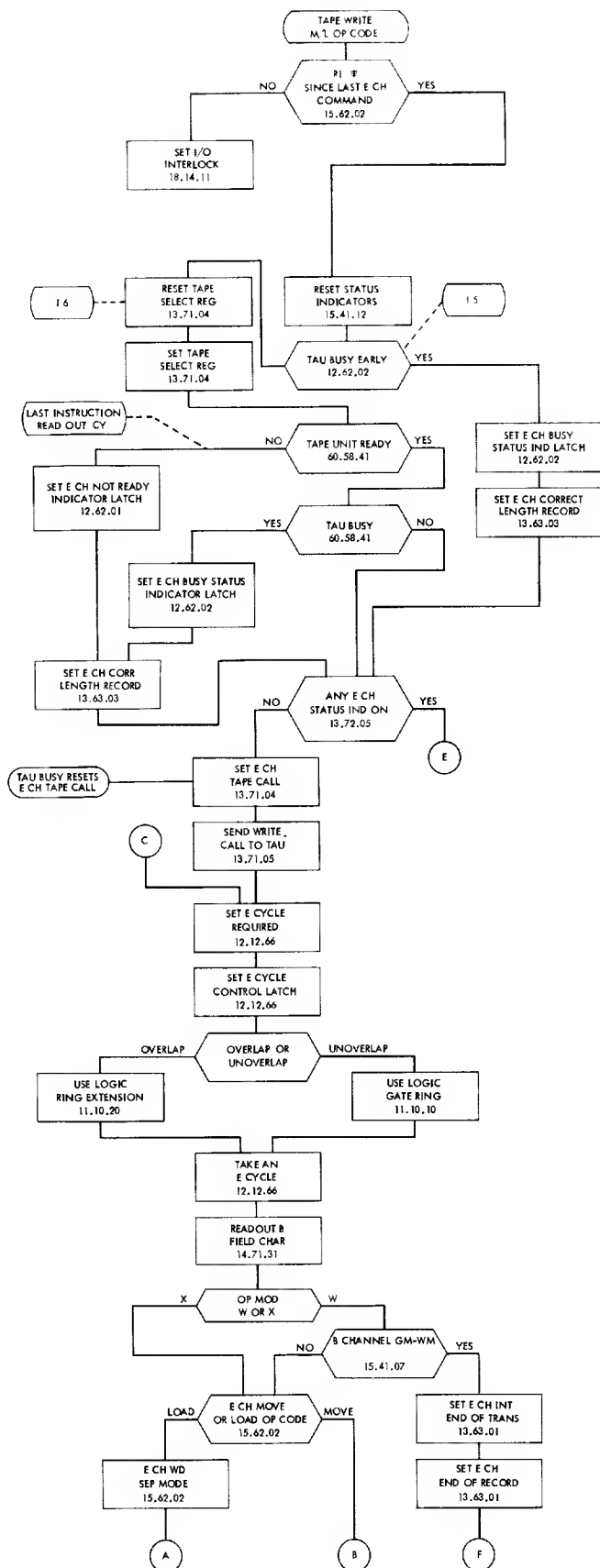


Figure 5.1-1A Tape Write Operation



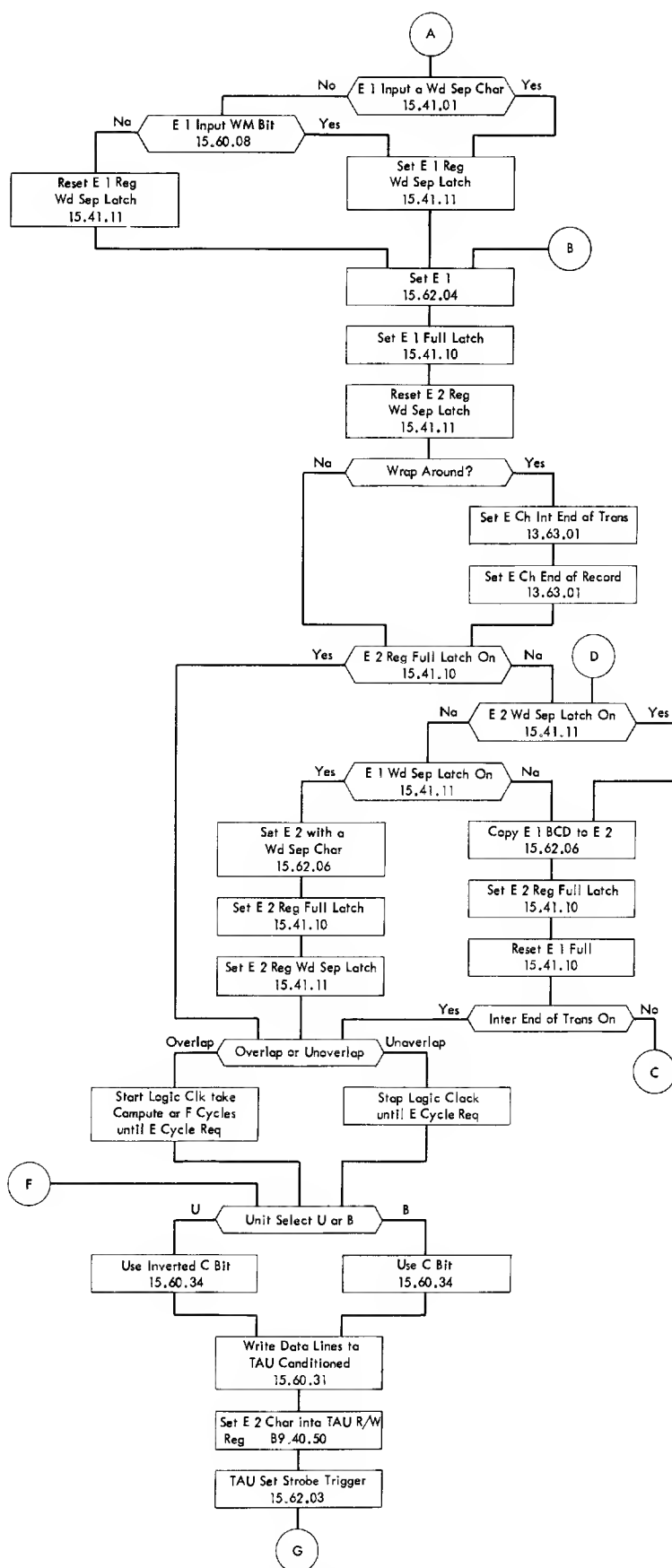


Figure 5.1-1B Tape Write Operation

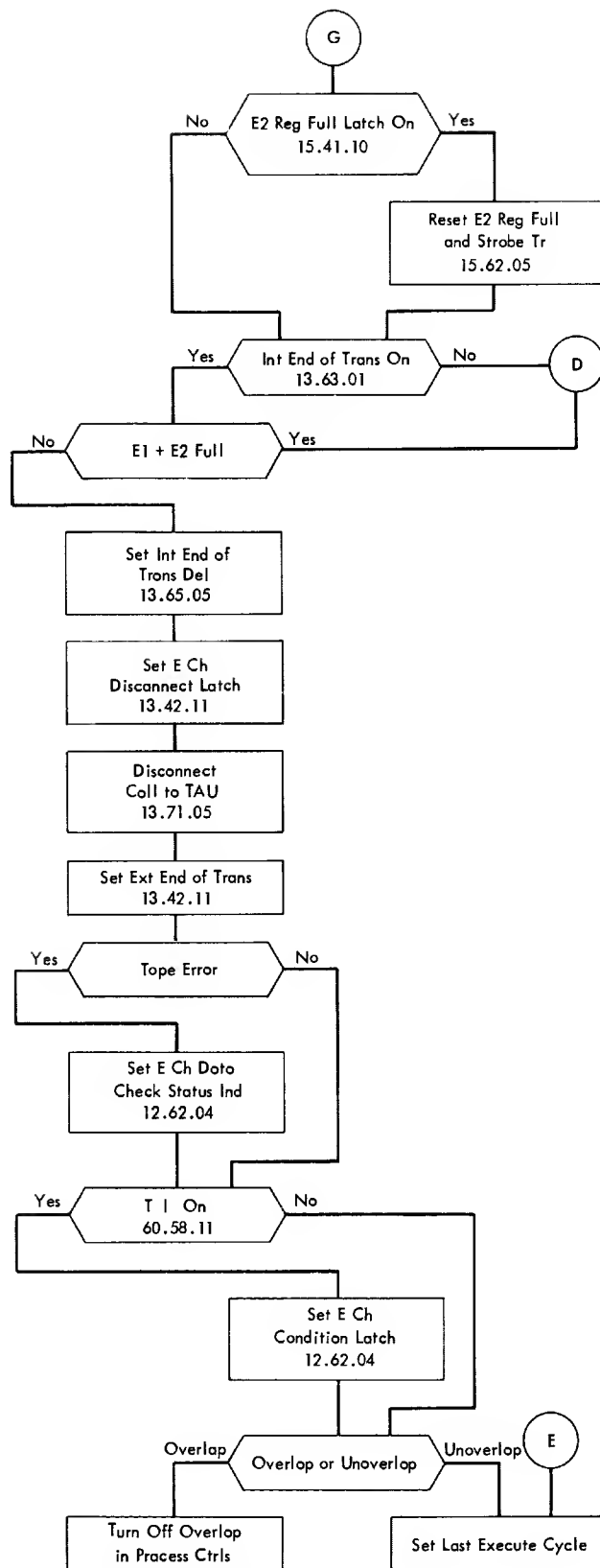


Figure 5.1-1C Tape Write Operation

is necessary to prevent selecting a new tape unit, when the TAU is still busy with the previous selection. An I/O interlock check must be made to insure that the status indicators have been tested since the last channel operation. The selected-tape unit must be ready for operation. These conditions, busy and not ready, cause status indicators to be turned ON, and no further operation for this instruction occurs. The program continues to the next instruction in sequence. If the status indicators are not turned ON, a write-call signal is sent to the TAU. When TAU accepts the character from the E2 register, it generates a write-strobe pulse that results in the transfer of another character from the E1 register into the E2 register. Another E-cycle transfers a new character from storage to the E1 register. This transfer of data, that the write strobe pulses initiate continues until a disconnect call is generated. If the computer is in overlap operation, compute cycles of F-channel cycles can occur between E-cycles.

The d-modifier of the instruction word controls the generation of the disconnect-call signal. The W-character d-modifier allows the disconnect to occur when a group-mark with a word-mark, or an end-of-storage condition is sensed. The X-character d-modifier allows the disconnect to occur only when an end-of-storage condition is sensed. When the TAU receives the disconnect call, it continues to read-check the characters just written on the tape. When the read-check of the write operation is complete, the TAU indicates the end-of-data transfer. Before the instruction operation is completed, a check for error conditions is performed. Status indicators are set, if the TAU detects an error or the end-of-tape marker is sensed by the selected-tape unit.

#### 5.1.02 Tape Read Operation (Figure 5.1-2)

The information from the selected-tape unit reads with or without word marks, and is stored in the location specified by the B-address.

The instruction format is the same as the tape-write format except for the d-modifier. The R-character d-modifier causes a normal read operation that ends when a group-mark with a word-mark is sensed. The S-character d-modifier causes a read operation that stops at the end-of-storage or the inter-record gap on the tape.

The selected-tape unit reads the data from tape, and transfers the data to the R/W register in the TAU. The output of the R/W register conditions the input to the E1 register. TAU indicates (read-strobe pulse) that a character is in the R/W register, and that the character is set in the E1 register with correct parity (parity is determined by the unit-select character of the X-field). The character transfers from the E1 register to the E2 register. When the E2 register indicates that it has a character, an E-cycle transfers the character to the B-field in storage.

The computer performs status checks and conditions control circuits before it generates the tape-read call signal. The early-busy check is made, and the selected-tape unit is tested for a ready condition. A check is made to insure that the status indicators have been tested since the last E-channel operation. If busy or not-ready conditions exist, the computer reads out the next instruction in sequence. If these conditions do not exist, a read-call signal is sent to the TAU. The selected-tape unit reads the tape, and transfers the characters to the read registers

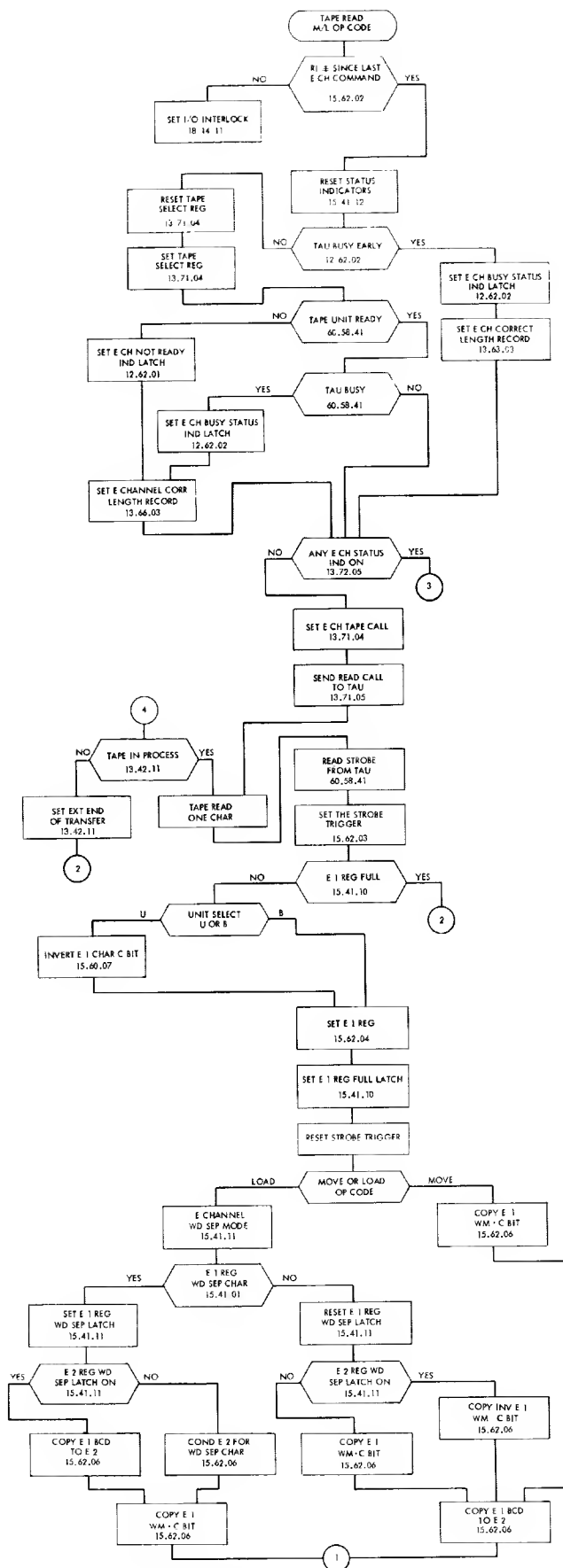


Figure 5.1-2A Tape Read Operation



in TAU. The read registers check the character and transfer it to the R/W register. When the R/W register has a character, TAU sends a read-strobe pulse to the computer. The character reads into the E1 register and then transfers to the E2 register. When the E2 register has a character, an E-cycle is required to transfer the character to storage. This transfer of data continues until the end of the read operation is sensed.

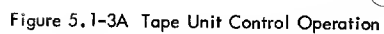
If the d-modifier is an R-character, the B-channel group-mark with a word-mark or the end-of-storage indication (wrap-around) causes the CPU to stop storing data. If the d-modifier is an S-character, the wrap-around condition causes the CPU to stop storing data. Wrap-around or group-mark with a word-mark conditions indicate the internal end-of-transfer. However, the selected-tape unit does not stop until the inter-record gap is sensed. TAU signals an external end-of-transfer when the tape unit senses the inter-record gap. Both the internal and external end-of-transfers are necessary to end the tape-read instruction operation. Before the operation ends, the CPU checks for error conditions. Status indicators set, if the TAU detects an error, if the end-of-tape marker is sensed by the selected-tape unit, or if the record that was just read is the wrong length.

### 5.1.03 Tape-Unit Control Operation (Figure 5.1-3)

The selected tape unit performs the operation specified by the d-modifier character.

<u>Instruction Format</u>	
<u>d-modifier character</u>	<u>Operation performed</u>
B	Backspace a tape record
E	Skip and blank tape
M	Write a tape mark
R	Rewind a tape reel
U	Rewind and unload a tape reel
A	Read-call for special operation

The unit-control instruction operation does not transfer data. The objective of the instruction is to send a control signal (determined by the d-modifier character) to the TAU to control the operation. Before the CPU generates the signal to TAU, it must check for the early-busy, not-ready, and I/O interlock conditions. For all unit control operations except Write a Tape Mark, the CPU finishes with the operation when the signal is sent to the TAU. The write-a-tape-mark operation is a one-character write operation. This instruction is the only one that can operate in the overlap mode. When the TAU receives a write-tape-mark call, a special one-character record writes on the tape. After the character is read-checked, TAU signals the CPU to end the operation. A check detects TAU errors and the end-of-tape marker. The A-character d-modifier with a special branch instruction (RIK) in maintenance programs checks the distance between the records on tape.



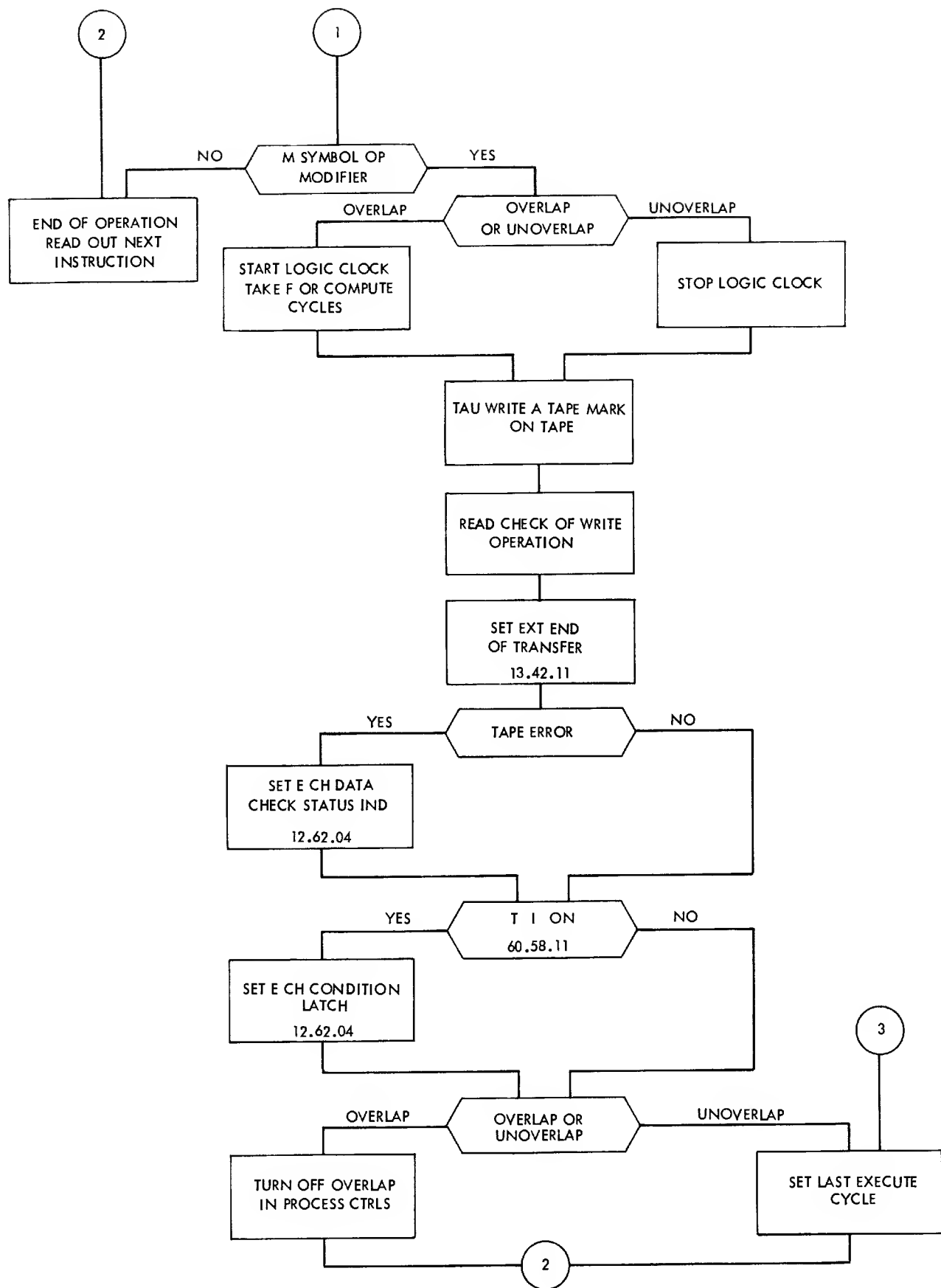


Figure 5.1-3B Tape Unit Control Operation



## 6.0.00 DISK-STORAGE CONTROL

### 6.1.00 DISK-STORAGE CONTROL INSTRUCTIONS

These instructions make it possible to seek a disk record, to read and to write a single record or a full track (with or without word mark), and to check for accurate results. The various parts of a disk-unit operation instruction are:

1. Mode of operation--move (M), or load (L). The move mode records data on, or reads data from, the disk unit in 7-bit coding (no word marks). The load mode records data on, or reads data from, the disk unit in 8-bit coding (word marks). Data must read from the disk unit in the same mode (move or load) in which it is written.
2. X-control field. This three-position field selects the transmission channel, the disk-storage unit, and the specific type of operation.
  - a. The hundreds position of the X-control field defines which data transmission channel is to be used, and whether the operation is to be performed with or without overlapping.
  - b. The tens position of the X-control field must contain an F-character to specify the disk-storage unit as the active input-output device.
  - c. The units position of the X-control field specifies the type of operation to be performed. The characters used in this position are: 0, seek operation; 1, single-record operation; 2, full-track operation; 3, write-check operation; and 4, write-address operation.
3. B-address. This address specifies the high-order core-storage position of the field that contains the 8-character disk-unit address and the record data.
4. d-modifier character. This position specifies a read (R) or write (W) operation.

The 8-character disk-unit address is used to locate the specific area of disk storage for the read or write operation. The format for the disk-unit address is as follows:

Format for the Disk-Unit Address

<u>Access Mechanism</u>	<u>Disk Unit No.</u>	<u>Disk-Unit Address</u>			<u>Fixed Character</u>
		<u>Disk Face</u>	<u>Track</u>	<u>Record</u>	
X 0, 1, 2	X 1, 1, 2, 3, 4	XX 00-99	XX 00-99	X 0-9	X Always Zero

#### 6.1.01 Seek-Disk Record M/L % F 0 (B) R

This Op code causes the selected access mechanism to locate the disk and track that are specified by the disk-unit address. Access time varies from 100 to 800 milli-

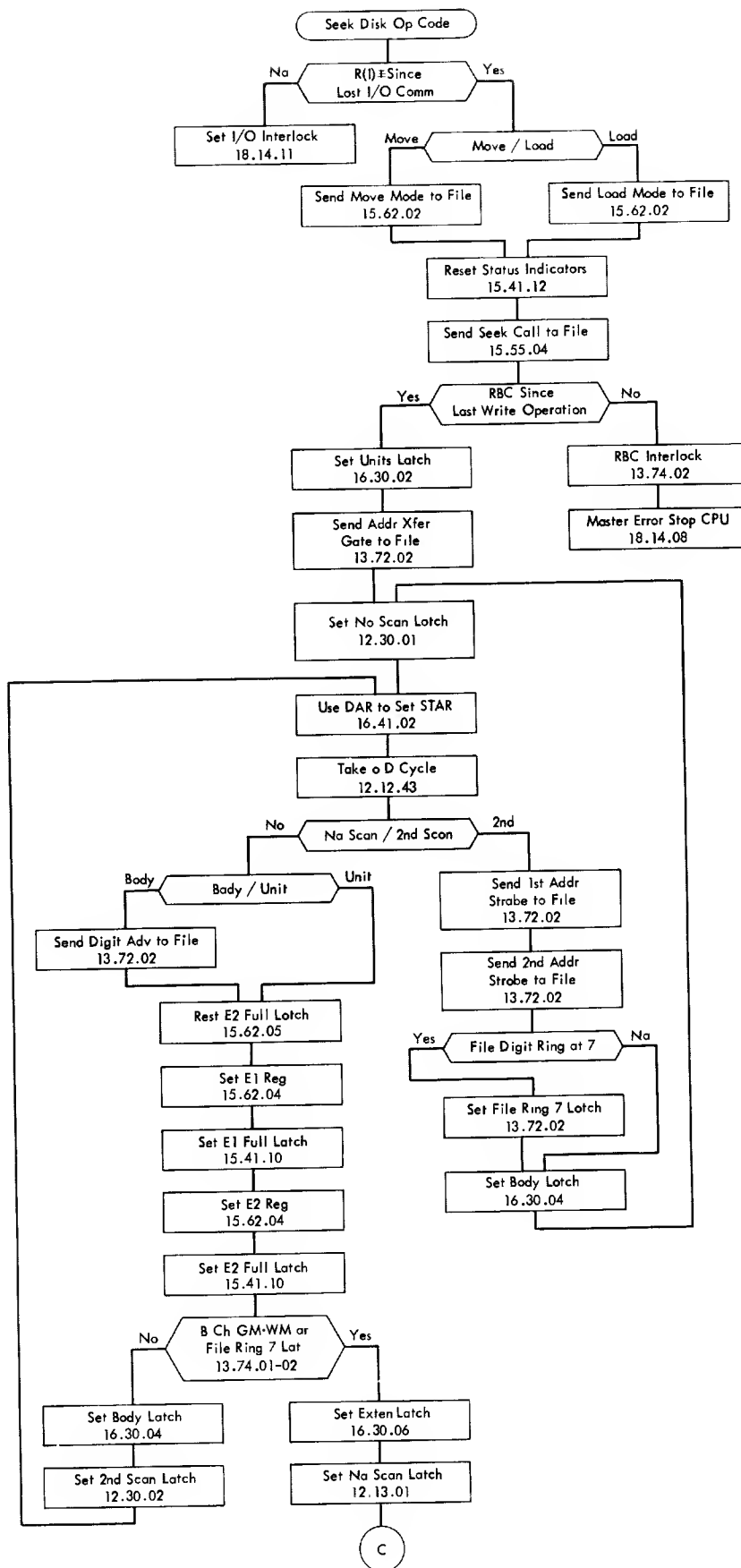


Figure 6.1-1A Disk Seek Operation

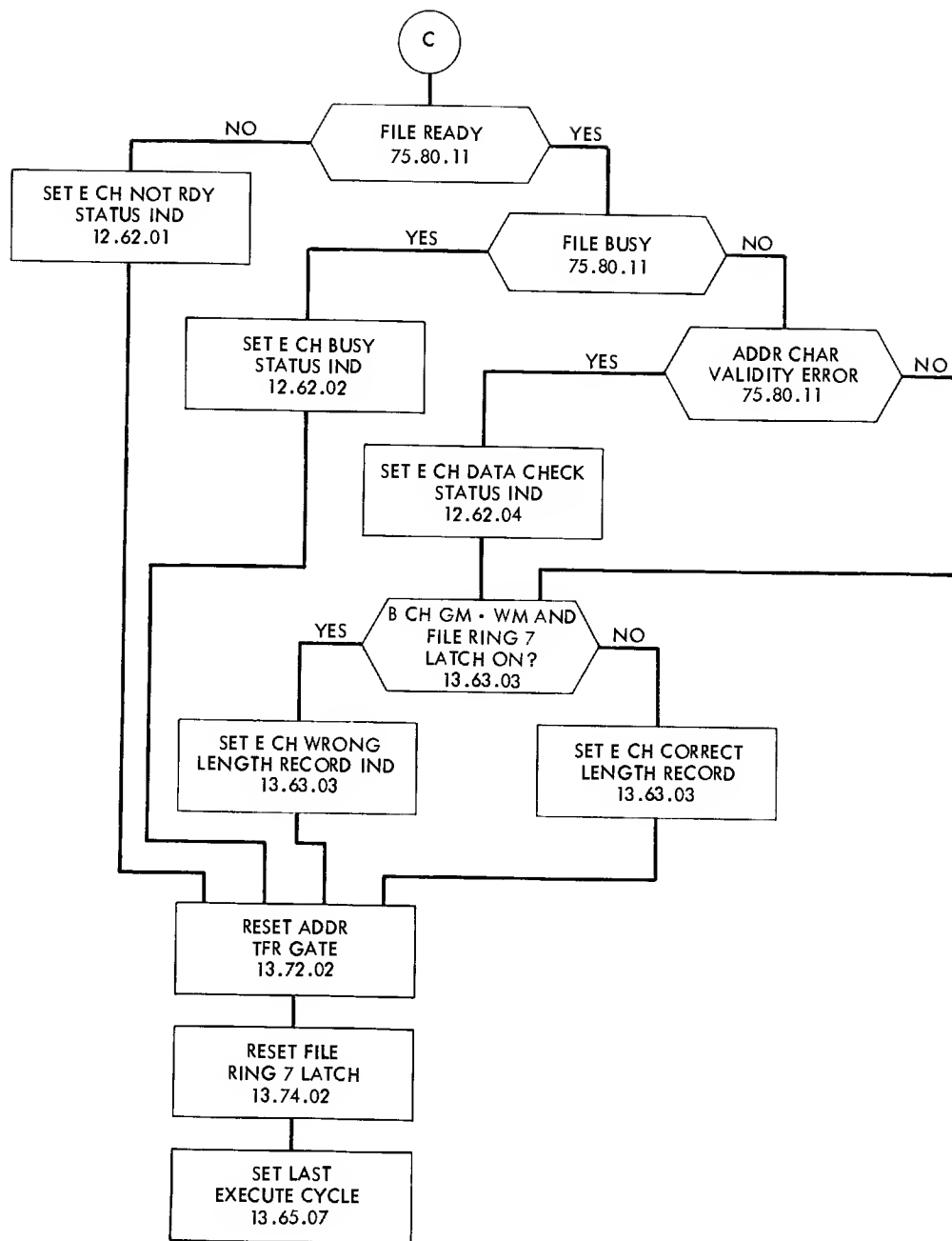
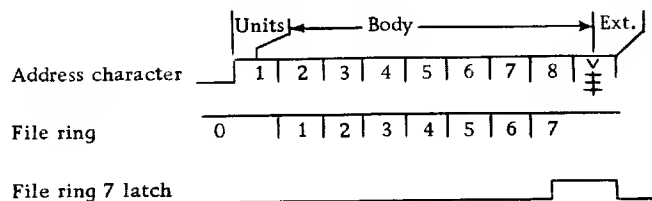


Figure 6. 1-1B Disk Seek Operation

seconds. The instruction sends the 8-character address to the file-control unit (FCU). After the FCU receives the address, the CPU continues on with the next instruction in sequence while the access mechanism moves to the desired disk and track.

The address must be 8-characters, followed by a group-mark with a word-mark. The file-digit ring is an eight-position ring that is used to count the characters in the address. The digit ring is at position 0 while the first character reads out and is sent to the FCU. The ring advances to position 1, and the second character reads out and is sent to file, etc.



It requires two D-cycles to read out each character and to send it to file. During the first D-cycle, the character reads out of storage and is gated to the E2 register. Also, during this cycle the B-channel is analyzed to determine whether the character is a GM·WM. If there is no GM·WM, and file-ring-7 latch is OFF, the character is sent to the FCU during the second D-cycle. If there is a GM·WM or the file-ring-7 latch is ON, the second D-cycle sets the status indicators with file-not-ready, busy, or with a validity-check error of a character that is sent to file. During this cycle, if the address is the correct length, the B-channel must be a GM·WM and file-ring-7 latch must be ON.

#### Operation (Figure 6.1-1)

During instruction read-out, after the I/O interlock check, the move or load mode is sent to the FCU. The E-channel controls and status latches are reset and a Seek Call is sent to file.

At this time, as in all file Op codes, the read-back check (RBC) interlock is checked to insure that a RBC was given as the first file Op after a file-write operation. If an RBC was given, the master error line is conditioned to stop the CPU.

The first cycle after instruction read-out is a D-cycle, with the no-scan latch ON. The DAR is used to read out the high-order (access-mechanism) character of the disk-unit address. The character is gated from the B-channel through the assembly to the E1 register. The E1 register full latch gates Set E2 Reg to set the character into the E2 register. There must not be a B-channel GM·WM at this time. The body latch and the 2nd-scan latch are set for the second D-cycle. The DAR still contains the address of the high-order character because the no-scan latch was set during the first D-cycle. The DAR is used to read the character out again. The 2nd-scan latch is ON during this D-cycle to modify the address in the DAR to the next position of the disk address.

A strobe pulse is sent to the file to indicate that a character is available on the E-channel and that the body and no-scan latches are set for the next D-cycle.

During this no-scan D-cycle, the body latch is ON to gate a digit-advance pulse to advance the file-digit ring to position 1.

The operation continues. The no-scan D-cycle is followed by a 2nd-scan D-cycle for each position of the file address. At the 2nd-scan D-cycle during which the 8th character of the file address is sent to file, file-ring-7 latch sets. The next character reads from storage, and should be a GM•WM.

During the final D-cycle, the status indicators are set. The presence of a GM•WM without the file-ring-7 latch or vice versa, results in a wrong-length address, and sets the E-ch wrong-length record indicator.

The Addr TFR gate and file-ring-7 latch are reset and last execute cycle is conditioned to initiate the next instruction.

#### 6.1.02 Disk-Address Transfer (Figure 6.1-2)

The address portion of the data in the B-field is transferred to the file before the actual data transfer during all read or write operations. After the first time the address transfers to the file, the various error conditions are checked, as well as the busy-status and ready-status of the file. If the file is ready for the read or write operation and if no errors are indicated, the address portion of data in the B-field transfers for the second time. This time the address compares with the indelible address read from the file location. After the second address transfer, the data reads from or writes on the file location.

The location of storage that the B-address of the instruction word specifies, contains the eight-character file address followed by a group-mark with a word-mark. The area of storage that follows the group-mark with a word-mark contains either the data to be written for a write operation, or the storage locations for the data to be read into on a read operation.

#### First Address Transfer

The CPU conditions the control lines to the file unit as soon as the instruction word is decoded. Move-mode or load-mode signals, full-track or single-record signals, and input-mode or output-mode signals are sent to the file. When the file unit receives these signals, control circuits are established to handle the address and data transfers. If the operation that is being performed is write-disk check (read-back check), a write check signal is sent to the file and the read-back check latch in the CPU is reset. The latch sets at the end of a normal write-disk operation. If any disk instruction, except write-disk check, is attempted with this latch on, the master error line stops the CPU. The CPU uses the D-address register, and executes D-cycles for the first address transfer. Two D-cycles are necessary to transfer one address character during the first address transfer. The no-scan and 2nd-scan latches control modification of the D-address register.

The no-scan D-cycle reads the disk-address characters out of the storage locations. The character is checked to determine if it is the group-mark with a word-mark. If the group-mark with a word-mark is sensed, the end of the first address transfer is initiated. The error lines are tested and the not-ready and busy conditions of the file are checked.

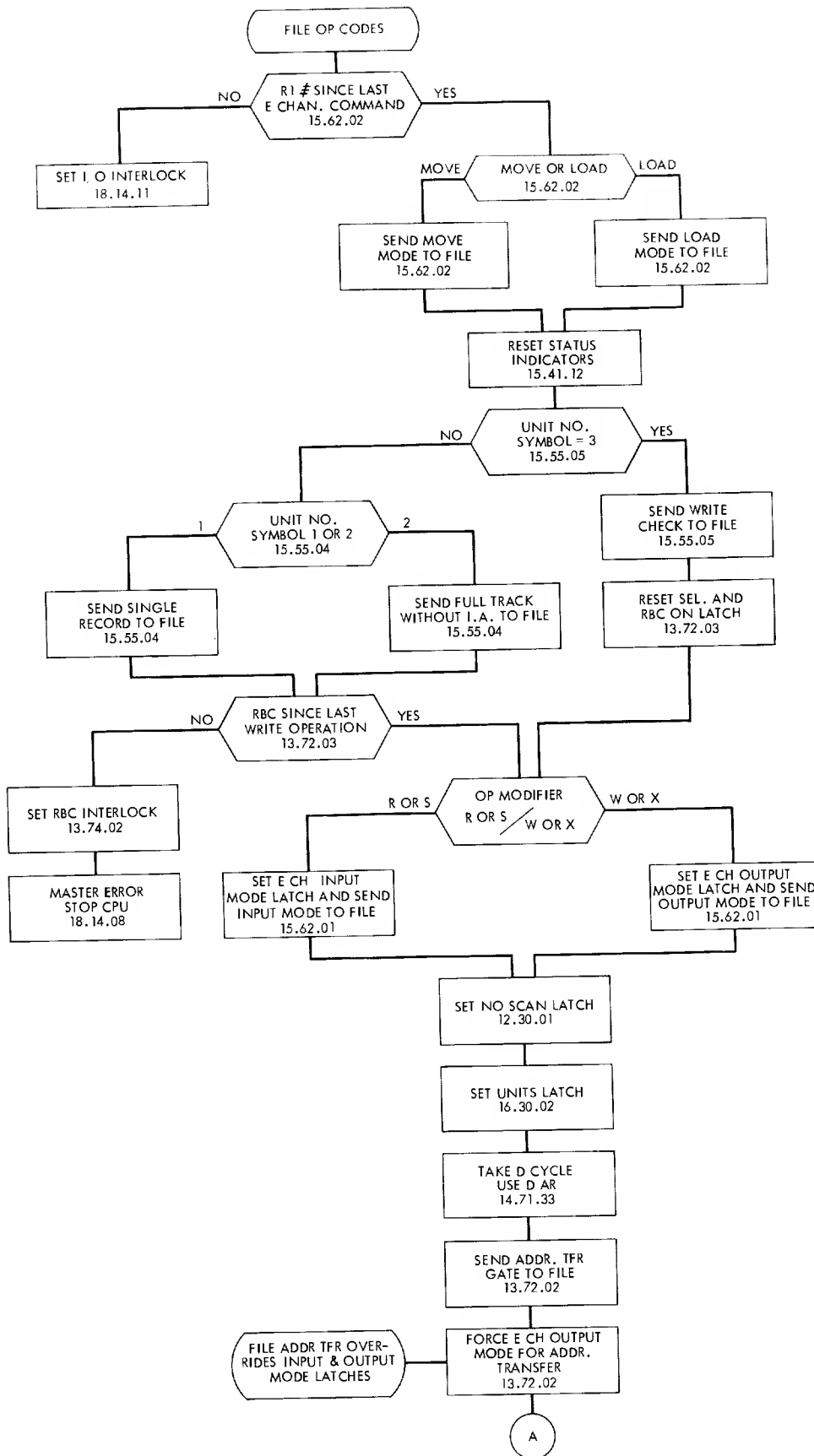


Figure 6.1-2A Disk Address Transfer

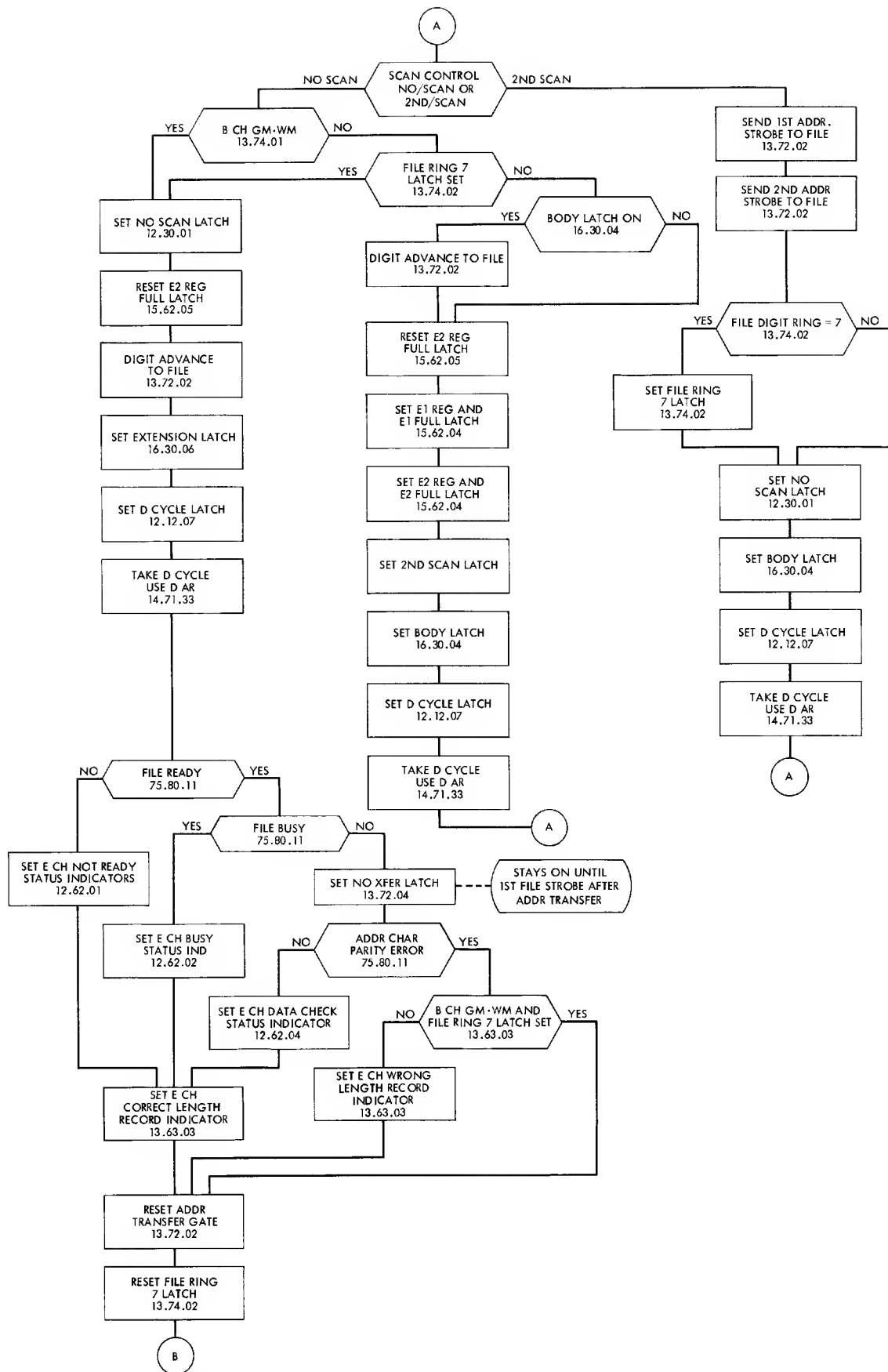


Figure 6.1-2B Disk Address Transfer

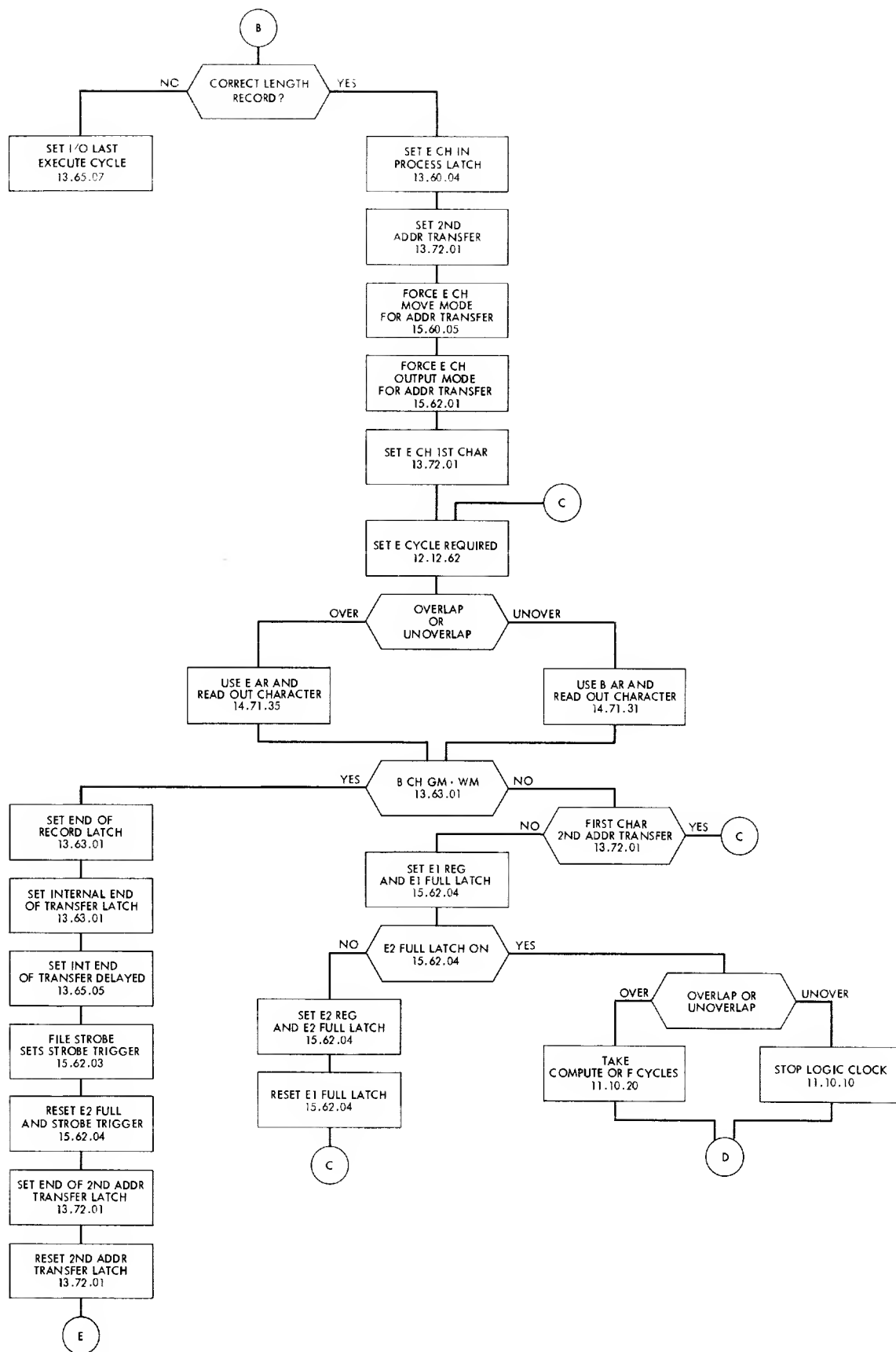


Figure 6.1-2C Disk Address Transfer



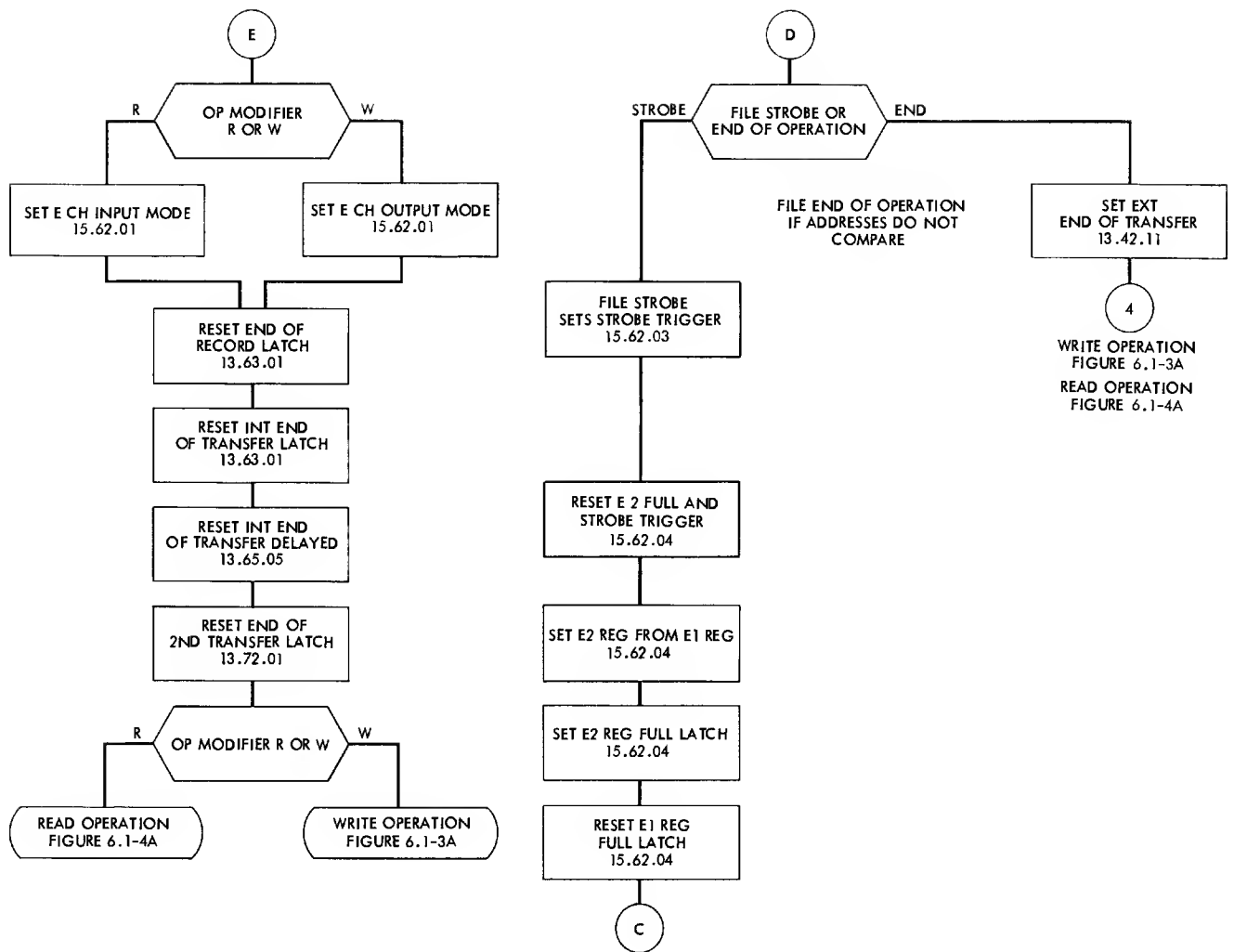


Figure 6.1-2D Disk Address Transfer

If the group-mark with a word-mark is not sensed, the file-ring-7 latch is checked. As each character in the body of the address field reads out of storage, the CPU sends a digit-advance pulse to the file. The digit-advance pulse steps the file ring. As the eighth and final character of the address field reads out of storage, the file ring steps to 7 and turns file-ring-7 latch ON in the CPU. This latch initiates the end of first address transfer in the same manner as sensing the group-mark with a word-mark. If the character that reads from storage is not a group-mark with a word-mark, and if the file ring is not at 7, the character sets into the E1 register and then transfers to the E2 register. The 2nd-scan latch sets and another D-cycle is taken.

The 2nd-scan D-cycle transfers the address character from the E2 register to the file. The CPU generates an address-strobe pulse to signal the file that a character is ready for transfer. The file samples the character in the E2 register each time an address-strobe pulse is received. During the 2nd-scan D-cycle, the D-address register is modified. The file ring in the file unit is sampled, and if it is equal to seven, the file-ring-7 latch in the CPU is conditioned. The no-scan latch turned ON and another D-cycle starts.

At the end of the first address transfer, the file is checked to determine whether it is ready and not busy. The characters in the address field are checked for parity, and the field is checked for proper length. If all of these conditions are satisfied (normal operation), the no-transfer latch turns ON and the second address transfer starts.

#### Second Address Transfer

The second address transfer is necessary to compare the address portion of the B-field to the indelible address that reads from the file. If the addresses do not compare, a file end-of-operation signal is generated and no data transfers. The second address transfer occurs in the move mode, and the E-channel is in output mode. These conditions are forced regardless of the instruction word. Only seven address characters transfer during the second address transfer. The first address character (access-arm identification) does not transfer. As the address characters read out of storage, they are sensed to determine if the group-mark with a word-mark has been reached. The group-mark with a word-mark starts the end of the second address transfer. This transfer is in an overlap or unoverlap mode. The characters read out, using the E- or B-address registers on E-cycles. An E-cycle is required when the E1 register is not full.

Strobe pulses, that are generated in the file, control address-character movement from storage to E1 and E2 registers. The strobe pulse indicates to the CPU that the file has accepted the E2 register character. The E1 register character transfers to the E2 register, and the next storage character reads to the E1 register. The first strobe pulse from the file resets the no-transfer latch.

When the character that reads out from storage is sensed as the group-mark with a word-mark, the end of the second address transfer starts. Control circuits for the second address transfer reset and the conditions that the instruction words call for are resumed. If the addresses compare, the CPU is now ready to transfer data to or to receive data from the file.

### 6.1.03 Disk Write (Figure 6.1-3)

There are three basic disk-write operations: write-a-single-record, write-a-full-track, and write-disk-check. Each operation can be given in either the move mode or the load mode, overlapped or unoverlapped. Data flow within the CPU is similar for each operation. Data transfers from the storage location to the E1 register. If the E2 register is not full, the data transfers from the E1 register to the E2 register. A character reads out of storage whenever the E1 register is not full.

#### Write a Single Record

The write-a-single-record operation causes a single record to write on disk storage from core storage. The data in core storage should fill the entire disk-unit sector area to prevent a CPU wrong-length record indication. As each character reads from storage, it is checked for the group-mark with a word-mark or wrap-around condition. The d-modifier of the instruction word determines which condition (GM·WM or wrap-around) causes the internal end-of-transfer. If the character is normal (not GM·WM or wrap-around) it is transferred to the E1 register, and then, to the E2 register, if the E2 register is not full. In the overlap mode, the computer takes compute or F-cycles while waiting for the file strobe. The file unit samples the output lines from the E2 register, and generates a file-strobe pulse to send to the CPU. The file-strobe pulse indicates that the file unit has accepted the character from the E2 register. Upon receiving the strobe pulse, the computer transfers a new character from storage.

The CPU senses the GM·WM or wrap-around condition after the last character of the record reads out of storage. The internal end-of-transfer is established and a disconnect signal is sent to the file. The file unit senses the end of the record and sends an end-of-operation signal to the CPU. The data is checked for proper parity and correct length, and Write a Single Record is concluded. If the write operation completes without errors, the read-back check latch is conditioned. The write-disk check operation must be the first file operation after the write operation or the CPU is stopped.

#### Write a Full Track

The write-a-full-track operation is similar to the write-a-single-record operation except for the end of the operation. The internal end-of-transfer is initiated, if the CPU senses the GM·WM or the wrap-around condition. The file generates the external end-of-transfer or file-end-of-operation when the end of the track is sensed. Each track contains five records.

#### Write Disk Check

The write-disk-check operation is similar to the write-a-single record operation except that no writing occurs in the file. The data transfers to the file and compares with the data that reads from the file. File-strobe pulses are generated when the file needs new information, and an end-of-operation is generated, if the characters do not compare. The no-compare condition causes the end of the write-disk-check operation and the E- or F-channel condition indicator is turned on.

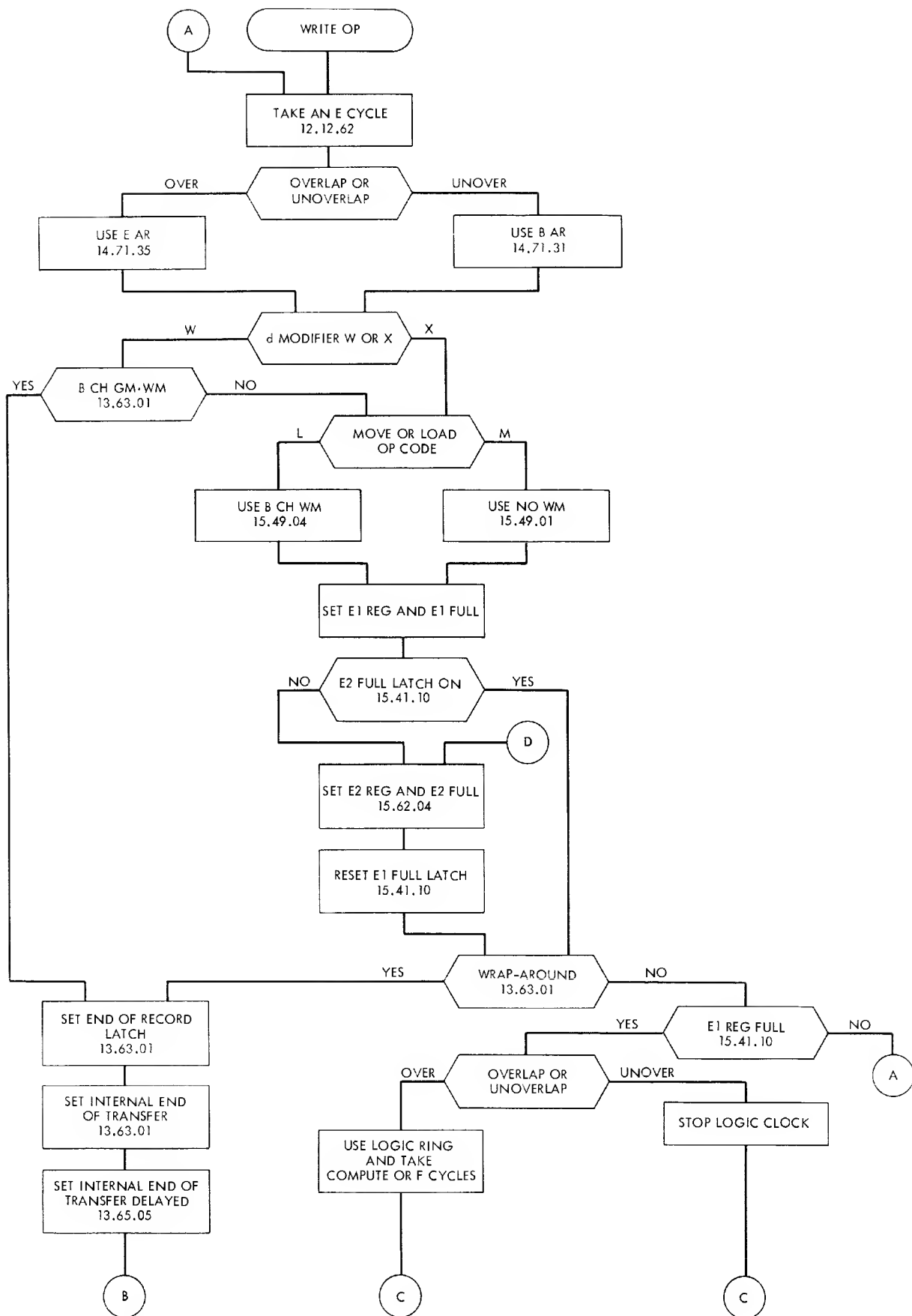
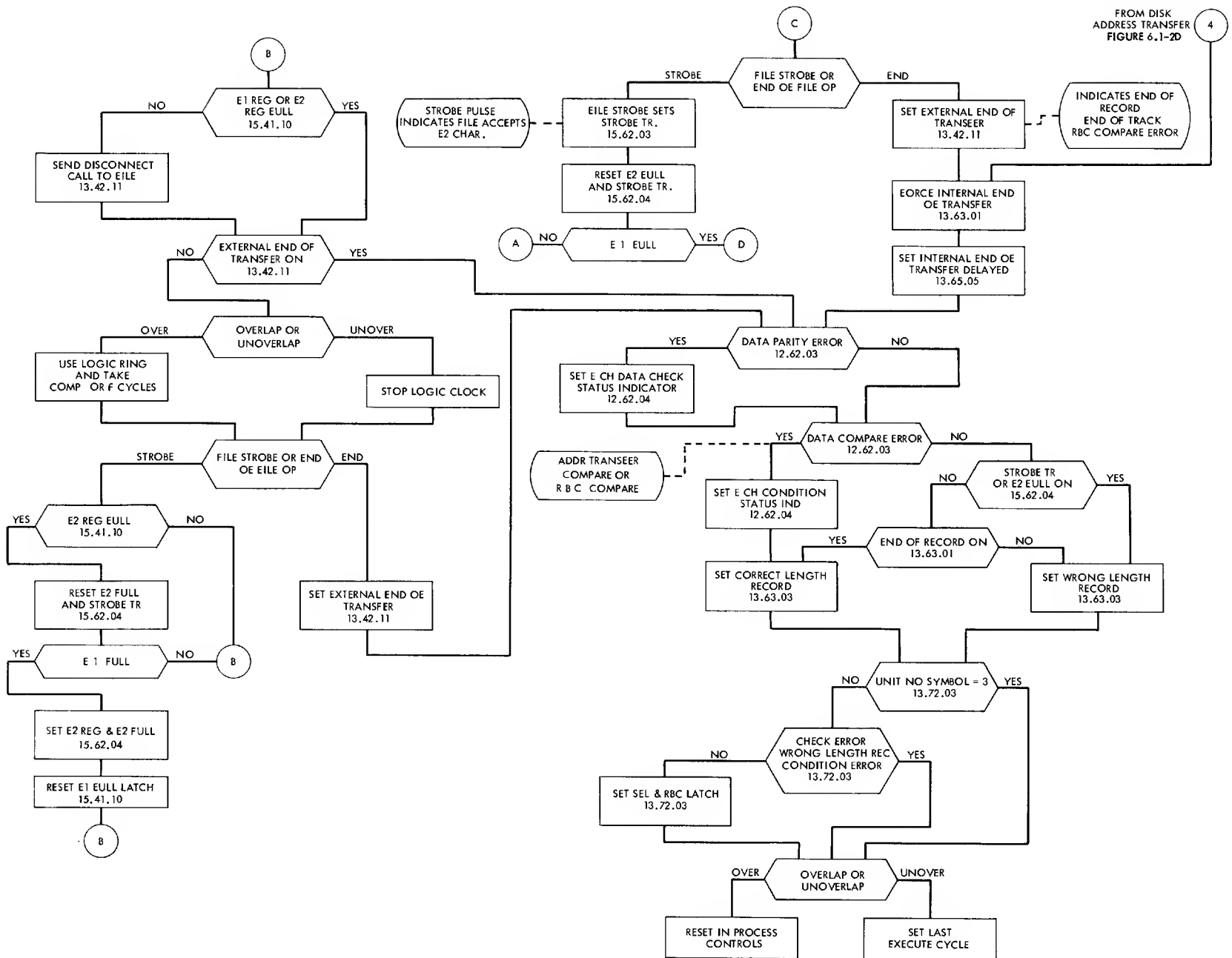


Figure 6.1-3A Write Operation

Figure 6.1-38 Write Operation



#### 6. 1. 04 Disk Read (Figure 6. 1-4)

##### General

The disk-read Op code causes the information that is available at the access arm in the file to read out and transfer to the CPU. The command is given in either the move or load mode. All information must be read in the same mode in which it is written.

During instruction read-out, the hundreds-position character of the X-control field defines the type of operation (overlap or unoverlap), as well as the data channel to use (E or F).

The tens position of the X-control field is always an F to specify a disk operation.

If the units position of the X-control field is a 1, a single record (200 characters) transfers to the CPU. A 2 in the units position of the X-control field causes an entire track (1000 characters) to read and transfer to the CPU.

The information that reads from the file is stored in the core-storage unit of the CPU starting at the high-order position that is specified by the B-field address. The d-modifier character must be an R to specify a read or input operation.

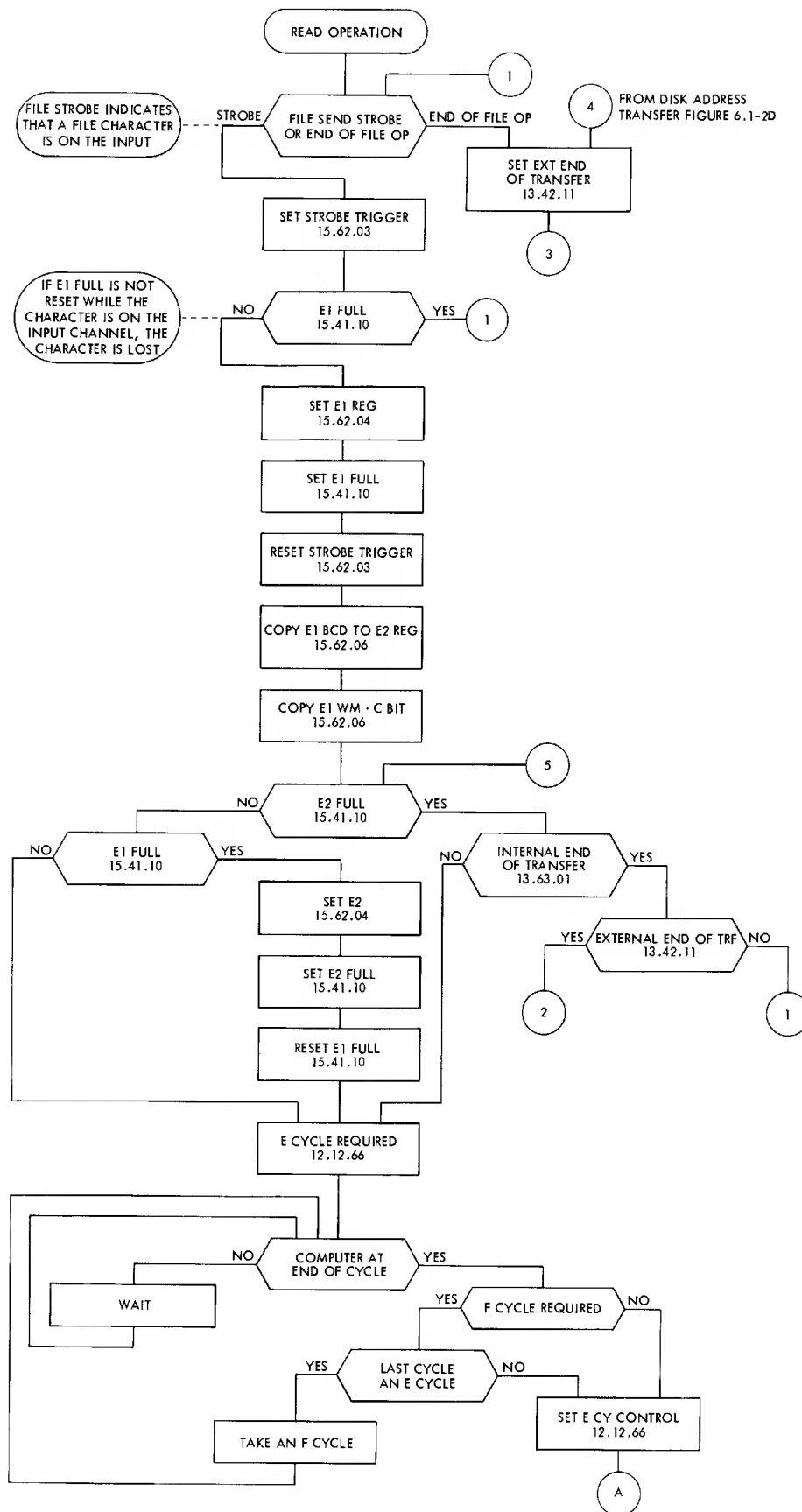
##### Read Operation

If the address transfer is made correctly (see section 6. 1. 02 Disk-Address Transfer), the transfer of file data to the CPU begins. Each time a character is sent to the CPU, a strobe pulse is sent to signify the presence of a character on the associated data channel (E or F). When the strobe is received, the character is placed in the proper data register (E1 or F1).

Assume that E-channel is used, the character transfers to the E2 register, if E2 is not full. When the E2 register is full, the CPU starts an E-cycle (overlap) or a B-cycle (unoverlap) to transfer the character to core storage. This sequence of transfers continues until the end of the record or the end of the track.

The file unit detects the end-of-record or end-of-track and generates a file-end-of-operation signal to send to the CPU. The file-end-of-operation signal sets the external end-of-transfer in the CPU. The CPU stores the character in E1 or E2 registers and senses a group-mark with a word-mark. The group-mark with a word-mark indication causes the end-of-record and internal end-of-transfer set. If the GM·WM is not sensed on this cycle, only the internal end-of-transfer sets, and a wrong-length record indication is established at status sample B-time. The wrong-length record also sets, if the GM·WM is sensed before external end-of-transfer.

With both internal and external transfers complete, the status sample B-pulse checks to determine whether a parity error occurred in the CPU or in the file unit.



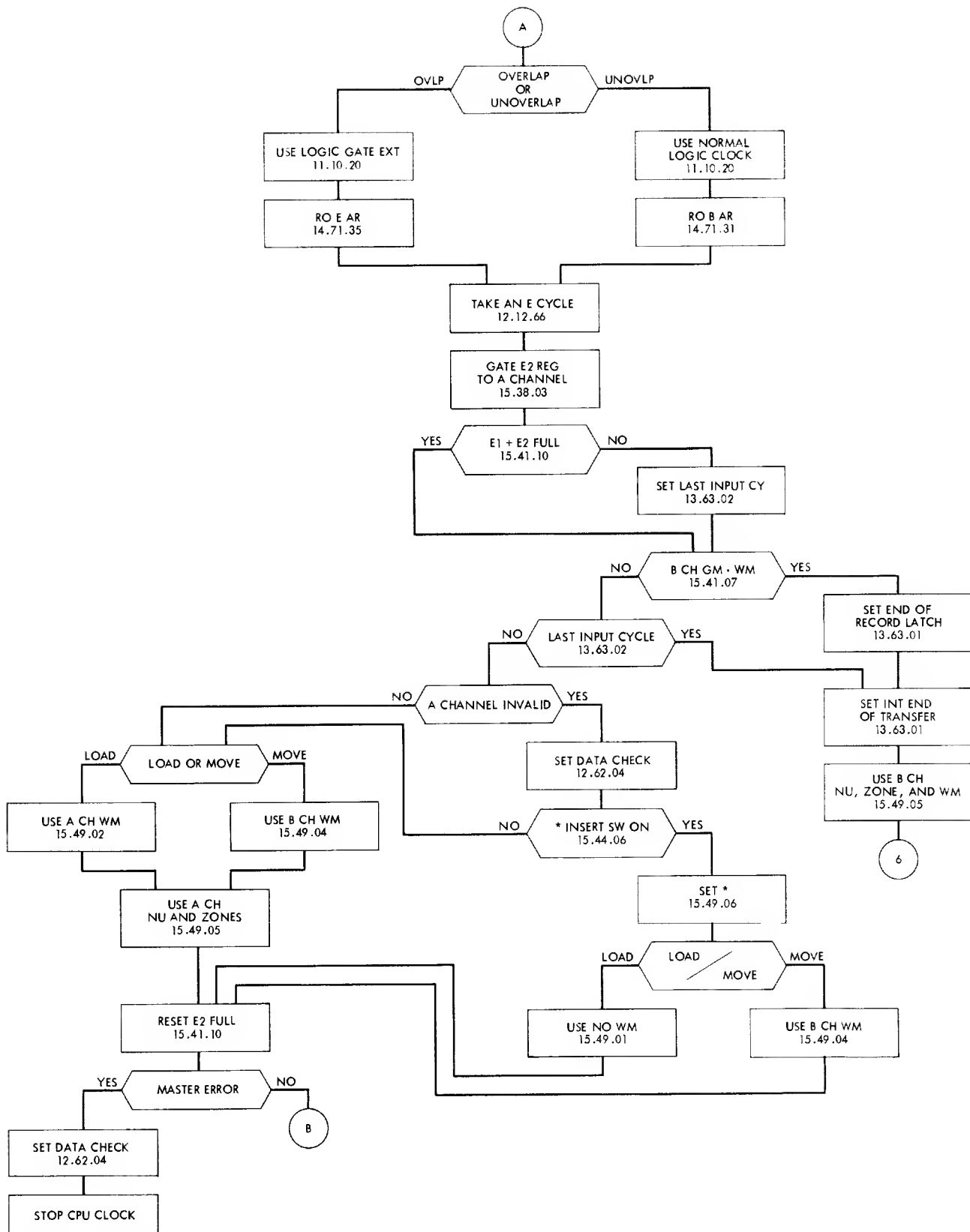


Figure 6.1-4B File Read Operation





If an error occurred, the data-check status indicator sets. The status sample B-pulse also checks for wrong-length record after the indicators are set, the CPU is released to continue with the stored program.

#### 6.1.05 Disk-Write Address

##### General

The write-address Op code allows the customer engineer to write a single, indelible address on a disk. The write-address key that is located on the disk-control-unit CE panel, must be ON before the instruction is executed.

##### Write-Address Operation (Figure 6.1-2)

The write-address operation starts with a first address transfer (see section 6.1.02 Disk Address Transfer). When the Addr TFR gate to the file drops after the first address transfer, the write gate is brought up to allow the file to write the address that transferred during second address transfer.

At the end of the second address transfer, unit number 4 prevents setting End of 2nd Addr TFR latch. This prevents the reset of End of Record, Int End of TFR, and Int End of TFR Delayed. The file sends End of Op after the address writes. This sets Ext End of TFR. Any parity error that is detected in the file, sets the data-check latch at status sample B-time. Status sample B also brings up the last execute cycle, if unoverlapped, to allow the program to proceed to the next instruction.

## 7.0.00 IBM 1415 CONSOLE AND INPUT-OUTPUT PRINTER

### General Description

Information is received from and sent to the IBM 1411 Central Processing Unit through the input-output printer. The printer can print all 64 IBM 1410-characters, 44 lower-case characters, and 20 upper-case characters. In addition, the printer can print an inverted circumflex or word-mark symbol (  $\vee$  ) over any character that has a word-mark associated with it. If the printer detects a parity error, the character in error is underscored (  $\_$  ).

Because it is possible to enter word marks from the printer, 128 different valid-bit configurations can enter the CPU from the console printer: 64 valid characters, plus the same 64 characters with word marks.

Three different types of print-outs can occur on the printer: stop print-out, display, and program print-out.

### Stop Print-Out

Whenever the CPU stops (either after an error occurs or at the end of a cycle on a single-cycle operation, or after a normal stop), the printer prints out the contents of the instruction address register, A-address register, B-address register, Op-register, Op-modifier register, A-data register, B-data channel, assembly channel, channel 1 unit-select register, channel 1 unit-number register, and (if two channels are in the system) channel 2 unit-select register and channel 2 unit-number register. The stop print-out can also be started by the start print switch.

### Display Print-Out

During a display operation from the console, the printer prints out the information stored in memory, starting at a previously entered address and going until the operator terminates the display.

### Programmed Print-Out

On a program print-out, the printer prints all the information that the programmer sends to it.

During address-set, display, alter, inquiry, and storage-scan operations, the printer enters information into the 1411. A five-digit address is keyed in on address set, display, and storage scan. In alter and inquiry, information is keyed into memory from the printer, including word marks, except on an inquiry-move operation.

The input-output printer consists of three main areas as far as the system is concerned: the keyboard for manual input, the solenoids for system output, and the cams and transmitting contacts. The printer has four basic cycles: print cycle, shift cycle, space cycle, and carriage-return cycle.

A print cycle can be initiated either from the keyboard or from the solenoids. Pressing a key to print a character, mechanically selects a particular combination of the seven mechanical-print-selection latches that govern the tilt and rotate motion of the print head. There is one solenoid for each print-selection latch. Therefore, by energizing the combination of solenoids, which is the same as the combination of latches selected by the key, the same print-selection latches are picked that were picked by pressing the key, and the same character prints on the paper. At least one selection latch must be picked to cause a print cycle.

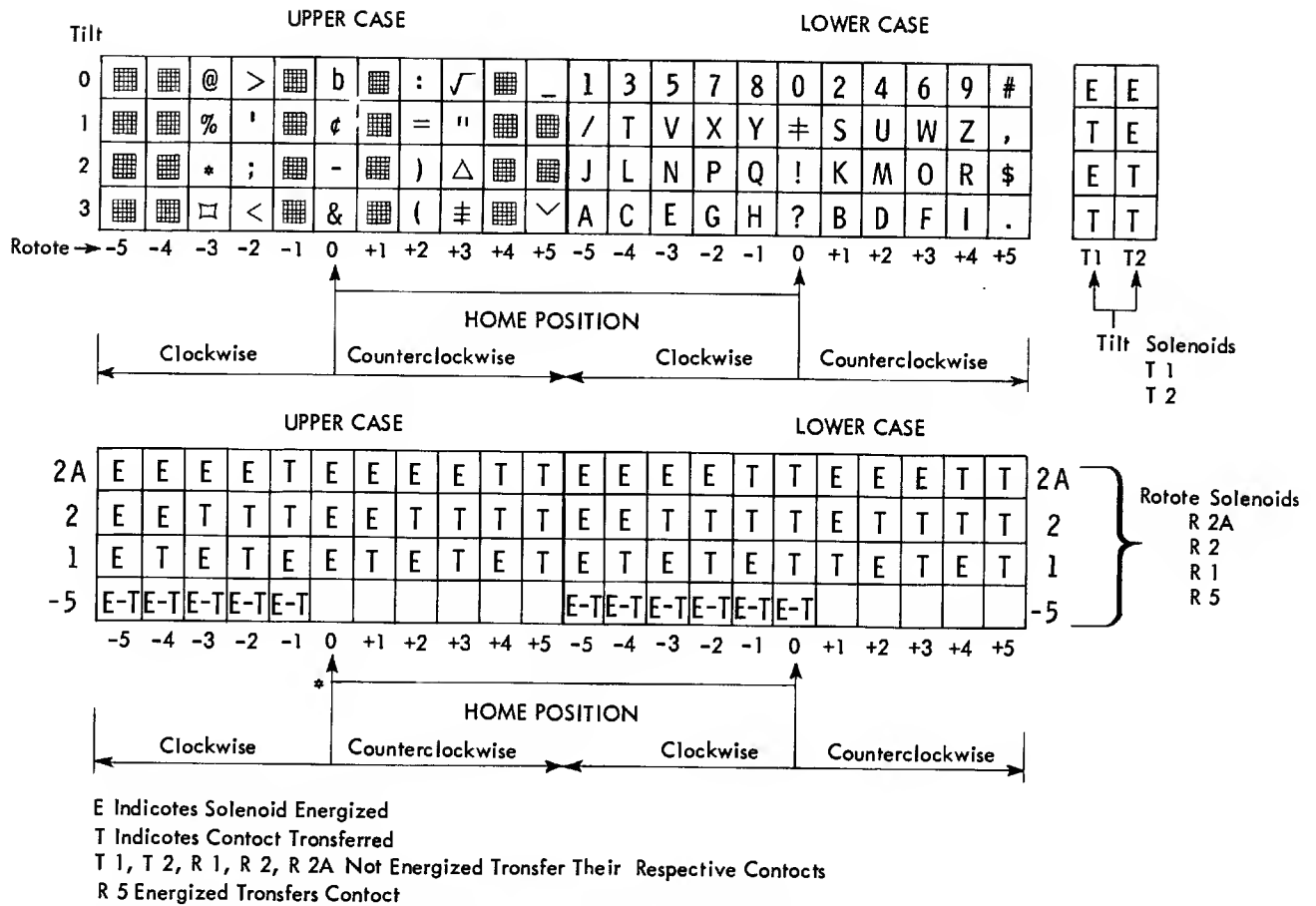
The seven selection latches (and solenoids) are R1, R2, R2A, and R5 (which govern the rotation of the print head), T1 and T2 (which govern the tilt motion of the print head), and CHK (which is used for an internal parity check), (see Figure 7.0-1). If no-tilt solenoids energize, the head tilts its maximum of three units. It tilts two units if T1 energizes, one unit if T2 energizes, and there is no tilt motion if both T1 and T2 solenoids energize. If no rotate solenoids energize, the head rotates five units counterclockwise. Energizing R1 rotates the head five units minus one unit, or four units counterclockwise.

In a similar manner, picking only R2A causes three counterclockwise units of rotation. Picking R2A and R1 causes two counterclockwise units of rotation. Picking R2A and R2 causes one unit of counterclockwise rotation. In order to obtain no rotation either R2, R2A, and R1 must all energize, or else only R5. Energizing R5 moves the head in a clockwise direction when any other rotate latch is selected. With R5 energized, the movement is one unit in the clockwise direction with R1 also energized; two units with R2A energized; three units with R2A and R1; four units with R2A and R2; and the full five units of clockwise rotation with all rotate solenoids energized.

Associated with each latch are four sets of transfer points that the cams operate when the print cycle takes place. The points for the R5 latch transfer when the solenoid energizes, but the points that are associated with all the other latches transfer when the solenoid is not energized.

A print cycle on the input-output printer operates in the following manner. The appropriate latches to print the desired character are selected either by key or by solenoid. After the latches are selected, the cycle clutch engages if this is the initial print cycle. If the printer has been printing and the latches for the next character are selected fast enough, the clutch remains engaged. The print clutch remains engaged, unless the printer fails to have its latches selected for the next character within five milliseconds of the finish of the previous character. The first point to close after the print-cycle shaft begins to rotate is the normally open point of cam C2 (Figure 7.0-2). The closing of this point signals that the drive to the solenoids should be dropped. The transfer points associated with the print selection transfer next. After these points transfer and the bounce is out of them, the normally open point of cam C1 closes. This signals that the information on the transmitting contacts (transfer points) is now available. Next the print-selection latches mechanically reset and the character prints on the paper. After the character is printed, the normally closed point on cam C2 is restored to normal. This signals that the printer is ready for its next character. A few milliseconds later, cam C1 is restored to normal.

## CHARACTER ARRANGEMENT



\*Manual depression of the b, ¢, - or & operates -5 and all contacts transfer

Figure 7.0-1 I/O Printer Character Arrangement

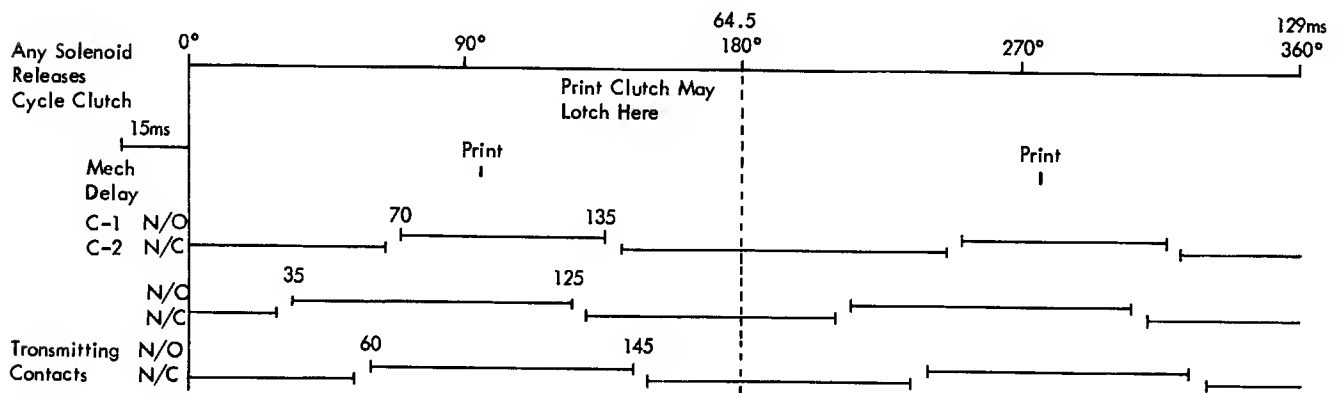


Figure 7.0-2 Print Cycle Timing

## Shift Cycle

Associated with a shift operation are a pair of transfer points that signal whether the printer is in upper case or in lower case, and a pair of cams, C3 and C4. C3 transfers on a lower-case to upper-case shift. C4 transfers on an upper-case to lower-case shift. The printer remains in upper case when controlled by the solenoids until a lower-case shift is called for. A shift operation can occur simultaneously with a backspace operation but not simultaneously with a print cycle.

After a shift cycle starts, the normally open point of C3 or C4 closes. This signals that the drive to the solenoid should be dropped. The transfer points then transfer to the new case. When C3 or C4 return to their normally closed state, the next cycle can start.

## Space Cycle

Pressing the word mark key (tab key), space bar, or energizing the space or backspace solenoids causes the printer to take a spacing cycle. A backspace operation cannot be initiated from the keyboard. As the spacing shaft begins to rotate, if the cycle is a space cycle, a pair of normally open points close to signal that a space operation is taking place. If the operation is caused by pressing the word-mark key, a normally open point closes to enter a word mark into the system. There is no such transmitting contact associated with backspace. After the transmitting contacts close, the normally open point of cam C5 closes to signal that the drive to the solenoids should be ended. After the operation is completed, the normally closed point of cam C5 again closes to signal that the printer is now ready for its next cycle. This signal is followed by the opening of the transmitting contacts.

## Carriage Return Cycle

A carriage return and index operations can be initiated only from the solenoid. The printer executes the index operation first, and then the print head returns to the left margin. As the cycle begins, the normally open point on the carriage return interlock closes to signal that the drive to the solenoid should be ended. When the carriage return ends, the normally closed point on the interlock is restored. It signals the next cycle to get underway. When the keyboard on the printer is locked, all character keys, the word-mark key, shift keys, and the space bar are inoperative. The keyboard is unlocked by energizing the lock solenoid. A set of transfer points on the printer indicates the state of the keyboard. The normally closed point on this keyboard-mode contact represents the unlocked state. The normally open point represents the locked state.

The last column contact closes near the end of the cycle occurring in the next-to-the-last column to signal that the next character will be in the last column.

Three inquiry keys are also housed on the I/O printer. The inquiry request key (carriage return key) is a normally open point that is used to request an inquiry. The inquiry release key (backspace key) is a normally open point that is used to signal the end of an inquiry message. An inquiry can be cancelled by pressing the inquiry cancel key (index key).

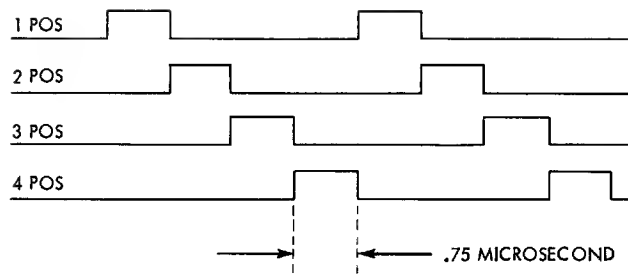


Figure 7.1-1 Console Clock

	Printout Ident		← I		A		R →		← A		A		R →		← B		A		R →		Op Code		Op Modifier		A Data Reg		B Channel		Assembly Reg		Unit Sel Reg		Ch 1	Unit Num Reg		Unit Sel Reg		Ch 2	Unit Num Reg				
X — Position	6	6	1	1	1	1	1	1	2	2	2	2	2	2	3	3	3	3	3	3	4	4	4	4	4	4	4	4	5	5	5	5	5	5	5	5	5	5	5	5	5		
Y — Position	5	6	1	2	3	4	5	6	1	2	3	4	5	6	1	2	3	4	5	6	1	2	3	4	5	6	1	2	3	4	5	6	1	2	3	4	5	6	1	2	3	4	5
Matrix Position	35	36	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33	34	35	36	37	38	39	40	

Figure 7.1-2 Stop Print-Out Format

MATRIX LOCATION	COORDINATES		FUNCTION (OUTPUT)	COORDINATES		FUNCTION (INPUT)
1	X1	Y1	T Th Bit I AR	X1A	Y1	Type T Th Bit
2	X1	Y2	Th Bit I AR	X1A	Y2	Type Th Bit
3	X1	Y3	H Bit I AR	X1A	Y3	Type H Bit
4	X1	Y4	T Bit I AR	X1A	Y4	Type T Bit
5	X1	Y5	U Bit I AR	X1A	Y5	Type U Bit
6	X1	Y6	Space	X1A	Y6	Carriage Return
7	X2	Y1	T Th Bit A AR			
8	X2	Y2	Th Bit A AR			
9	X2	Y3	H Bit A AR			
10	X2	Y4	T Bit A AR			
11	X2	Y5	U Bit A AR			
12	X2	Y6	Space			
13	X3	Y1	T Th Bit B AR			
14	X3	Y2	Th Bit B AR			
15	X3	Y3	H Bit B AR			
16	X3	Y4	T Bit B AR			
17	X3	Y5	U Bit B AR			
18	X3	Y6	Space			
19	X4	Y1	Op Register			
20	X4	Y2	Op Modifier Register			
21	X4	Y3	Space			
22	X4	Y4	A Data Register			
23	X4	Y5	B Data Channel			
24	X4	Y6	Assembly			
25	X5	Y1	Space			
26	X5	Y2	Unit Select Channel 1			
27	X5	Y3	Unit Number Channel 1			
28	X5	Y4	Unit Select Channel 2			
29	X5	Y5	Unit Number Channel 2			
30	X5	Y6	Special Character			
31	X6	Y1	Space			
32	X6	Y2	I/O Cycles No Word Mark Control			
33	X6	Y3	I/O Cycles Word Mark Control			
34	X6	Y4	Carriage Return			
35	X6	Y5	Special Character			
36	X6	Y6	Space			
Home	X6	—	—			

Figure 7.1-3 Console Matrix Control

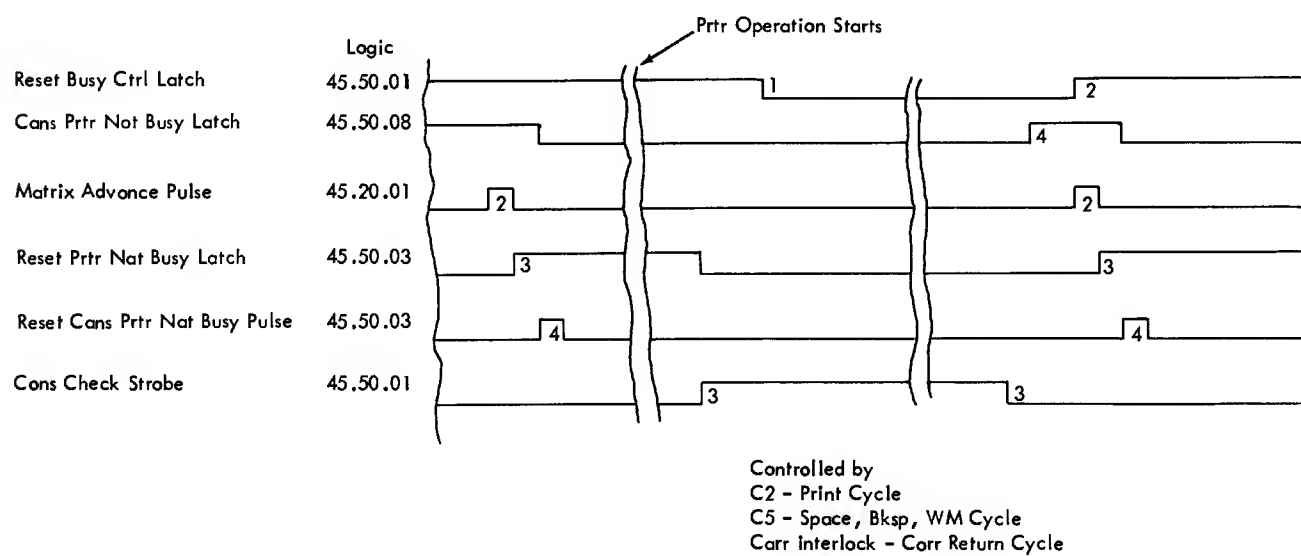


Figure 7.1-4 Matrix Advance Control



## 7.1.00 CONSOLE FUNCTIONAL UNITS

### 7.1.01 Console Clock (Figure 7.1-1)

The console clock-timing pulses set up all printer and console controls. The CPU oscillator drives this clock. It runs continuously and is completely independent of any other clock in the system. It provides four .75-microsecond pulses that are called console clock one position, console clock two position, console clock three position, and console clock four position. The program reset line resets the clock to the 1 position.

### 7.1.02 Console Cycle-Control Matrix

The console cycle-control matrix-ring gates out information from the central processing unit to the console printer, gates in information to the CPU from the console, and sets up controls to space or return the carriage, or to print a special character.

All console operations are controlled by the console matrix. The matrix consists of seven X-triggers and six Y-triggers. The X-triggers are called X1A, X1, X2, X3, X4, X5, and X6. The Y-triggers are called Y1, Y2, Y3, Y4, Y5, and Y6. The combination of one X-trigger and one Y-trigger turned ON can:

1. set up a gate to allow a unit, or a position of a unit, in the CPU to be read out and printed by the console printer.
2. set up a gate to permit console information to enter the CPU.
3. set up controls to space or return the carriage or to print a special character.

The console matrix remains in home position (X6 trigger ON, all Y-triggers OFF) between console operations.

The progress of the ring and steps that it takes varies with the type of console operation being performed. Figure 7.1-2 illustrates the steps that are taken and the function that takes place during each matrix step in a stop print-out operation. Figure 7.1-3 illustrates the function that takes place during each of the 36 matrix positions. The progress of the ring and the steps it uses again varies with the type of console operation being performed.

### 7.1.03 Console Cycle-Control Matrix-Advance

The console cycle-control matrix advances during a print-out operation with a console clock two-position when the Cons Prtr Not-Busy latch is ON. Because the matrix advances much faster than the mechanical operation completes, it is necessary to interlock the advance pulses. This is done by resetting the Cons Prtr Not-Busy latch after advancing the matrix, and by keeping the latch OFF until the operation completes (Figure 7.1-4).

#### 7.1.04 Character-Print Operation (Figure 7.1-5)

##### Output

The first operation of every console operation is printing the print-out identification character. Before an operation starts, the matrix is in the home position, the keyboard is locked (Prtr Locked Cnd Proceed trigger ON), and the Cons Prtr Not-Busy latch is ON. The type of operation gates the matrix to position 30 or 35. The next matrix advance sets the matrix triggers. The console-matrix-not-at-home conditions the take-console-printer-cycle gate, if the matrix-controlled function is not a manual operation (keyboard unlocked). Take Console Printer Cycle gates on the Reset Printer Not-Busy latch. This resets the Cons Prtr Not-Busy latch to prevent the matrix from advancing until the printer cycle completes.

For a character-print operation, the matrix' position gates the character to the console output bit lines. If the character has a WM, a WM routine is necessary to print the WM with the character under it. If there is no WM, the Cons Char Ctrl latch sets.

The console output lines are analyzed to determine whether the character is an upper-case (UC) or lower-case (LC) character. A bit configuration of  $\bar{8} \cdot \bar{4} \cdot \bar{2} \cdot \bar{1}$  or  $8 \cdot 4$  indicates a UC character. The LC and UC contacts indicate the shift that the printer is already in. The Cons Char Ctrl latch indicates that the character on the output lines is to be printed, and it allows the output bit lines to energize the UC or LC shift solenoids if a shift is required.

If a shift cycle is taken, the console printer shift-complete line is deconditioned until the shift cycle completes. The shift-complete line gates the setting of the solenoid-driver strobe latch that along with the console character control latch gates the output lines to energize the rotate, tilt and check solenoids.

Any solenoid that is energized trips the print clutch to start the print cycle. When C2 makes at 35°, the console check strobe sets. This resets the solenoid-driver strobe to de-energize the solenoids.

When C1 makes at 70° with the console character control latch ON, the console-printer-strobe trigger sets to test the parity of the character being printed. If the number of contacts that transfer is odd, there should not be a word-mark. If the number of contacts that transfer is even, there must be a word-mark. When these conditions are not satisfied, the console output error latch sets. This causes an error routine to underscore the erroneous character.

The check solenoid is the only solenoid that is energized when a period prints (WM C B A 8  $\bar{4}$  2 1). When a period prints with a WM over it (WM C B A 8  $\bar{4}$  2 1) the C-bit normally prevents energizing the check solenoid, but at least one solenoid must be energized to start a print cycle. When this character appears on the output lines, it gates the check solenoid to force the print cycle, and sets the word-mark period latch. With only the check solenoid energized, the number of contacts that transfer is odd (1). Word-Mark Period Latch ON prevents the console printer strobe from testing the contacts for parity, because an odd number of contacts that transfer combined with Word Mark would turn on the output error latch.

When cam C2 opens toward the end of the print cycle, it turns OFF the console check strobe latch that starts the console cycle-latch-set pulse to set the console printer-not-busy latch. The console cycle-latch-reset resets the character control latch. With the console printer-not-busy latch ON, the console matrix can now advance, and the printer takes its next cycle.

#### Input

With input operations, the matrix controls when the keyboard unlocks. Pressing a character key starts a print cycle during which the latch contacts transfer and are decoded into BCD (40.30.01). When the printer strobe trigger sets ON, the output of the latch contacts is gated to the CPU on the E-channel. The matrix may or may not advance depending on the operation.

#### 7.1.05 Space Operation (Figure 7.1-5)

#### Output

After every print-out identification character, the printer spaces before the first character of the line. At the completion of the character print, the console printer-not-busy latch advances the matrix to position 36 or 31, depending on the operation. Positions 31 and 36 gate the console space function that sets the function control latch and prevents setting the console character control latch. Cons Char Ctrl latch not ON prevents energizing any rotate and tilt solenoids. The function control latch and the solenoid driver strobe energize the space solenoid to start a space cycle.

During the space cycle, C5 sets the check strobe. The space N/O contact sets Prtr Strobe and prevents setting the output error latch. Again when the check strobe resets, the Cons Prtr-Not-Busy latch sets to allow the matrix to advance.

#### Input

During the input mode, the space key is pressed to enter a blank into the CPU. During the space cycle the space N/O contact makes to gate a C-bit to the CPU.

#### 7.1.06 Output Error Routine

The output error routine causes the printer to backspace and to print an underscore under the erroneous character before the matrix advances to the next position.

When the parity of the character is incorrect, the printer strobe sets the output error latch. At the end of the print cycle, the output error latch combines with the Cons Char Ctrl latch that is still ON, to set the Cons Bksp Ctrl latch. The Bksp Ctrl latch gates the printer to UC, if required, because the underscore is a UC character. The Cons Char Ctrl latch resets to prevent gating the output lines to the rotate and tilt solenoids during the backspace and underscore print cycle.

Because a shift and backspace cycle can take place simultaneously, Bksp Ctrl keeps Shift Complete up to set the solenoid driver strobe. The solenoid driver strobe and the Bksp Ctrl latch gate the Bksp Solenoid to start the Bksp Cycle. C5 again is used to control the check strobe. When the check strobe resets, the output

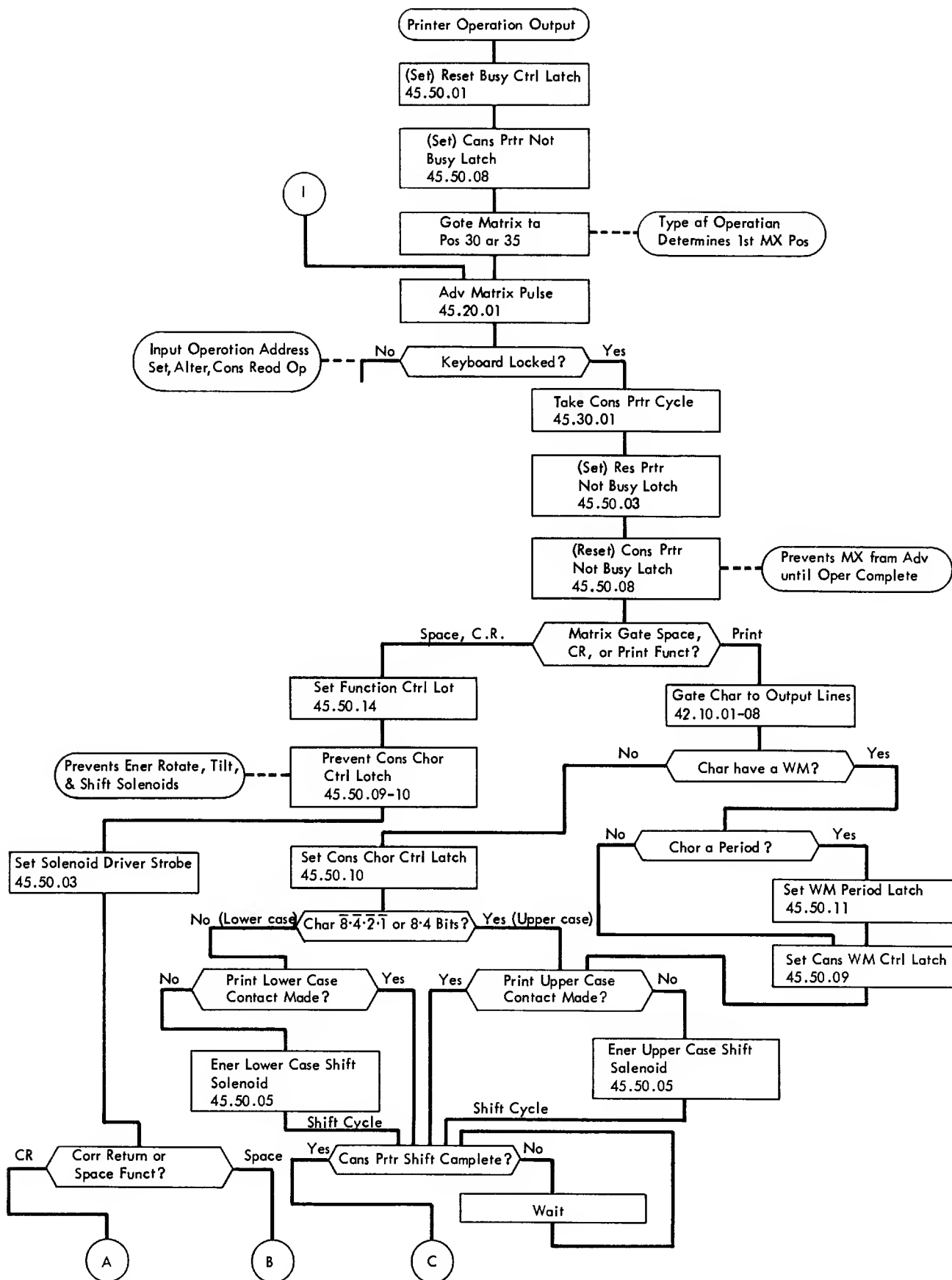


Figure 7.1-5A Printer Operation (Output)

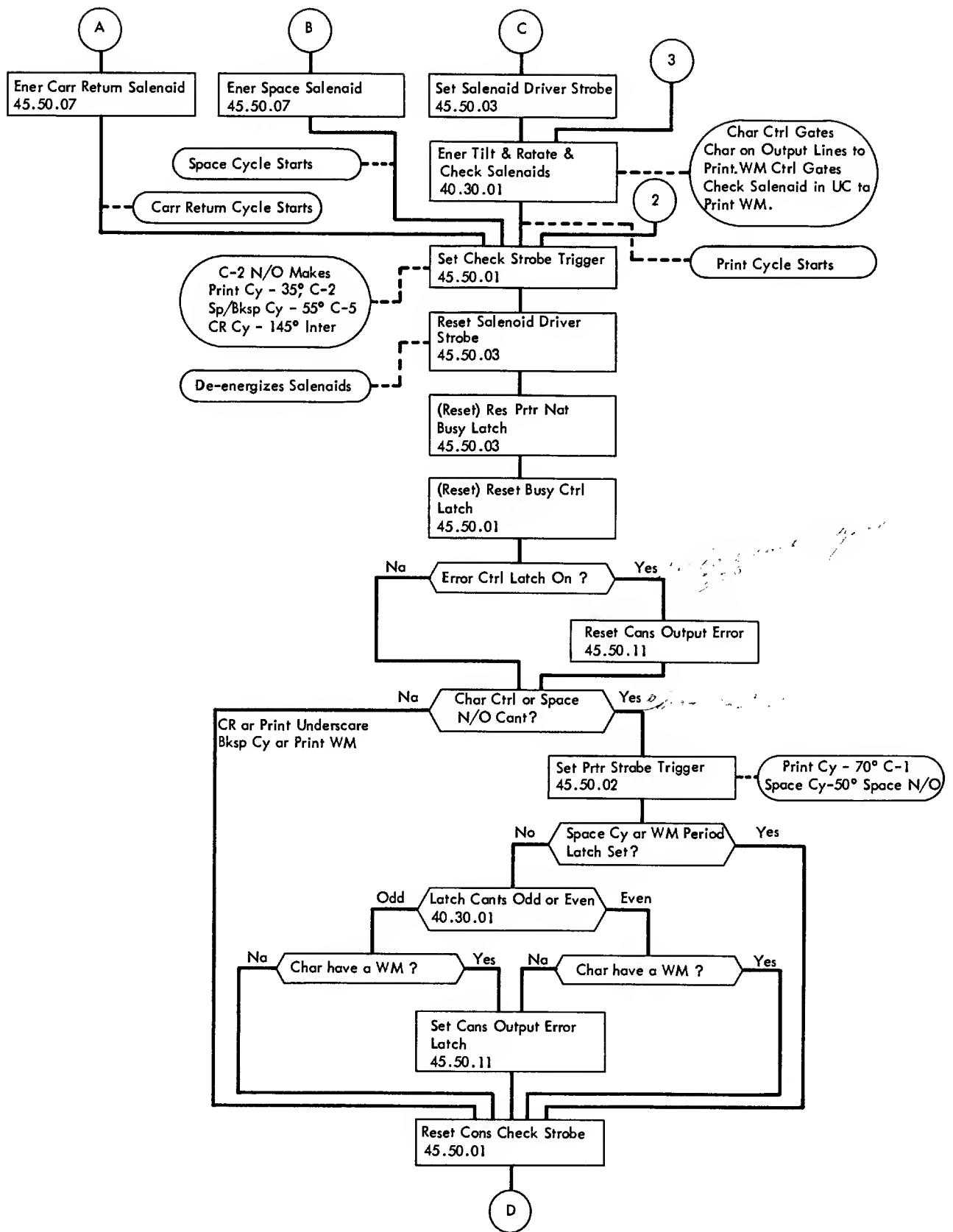


Figure 7.1-5B Printer Operation (Output)

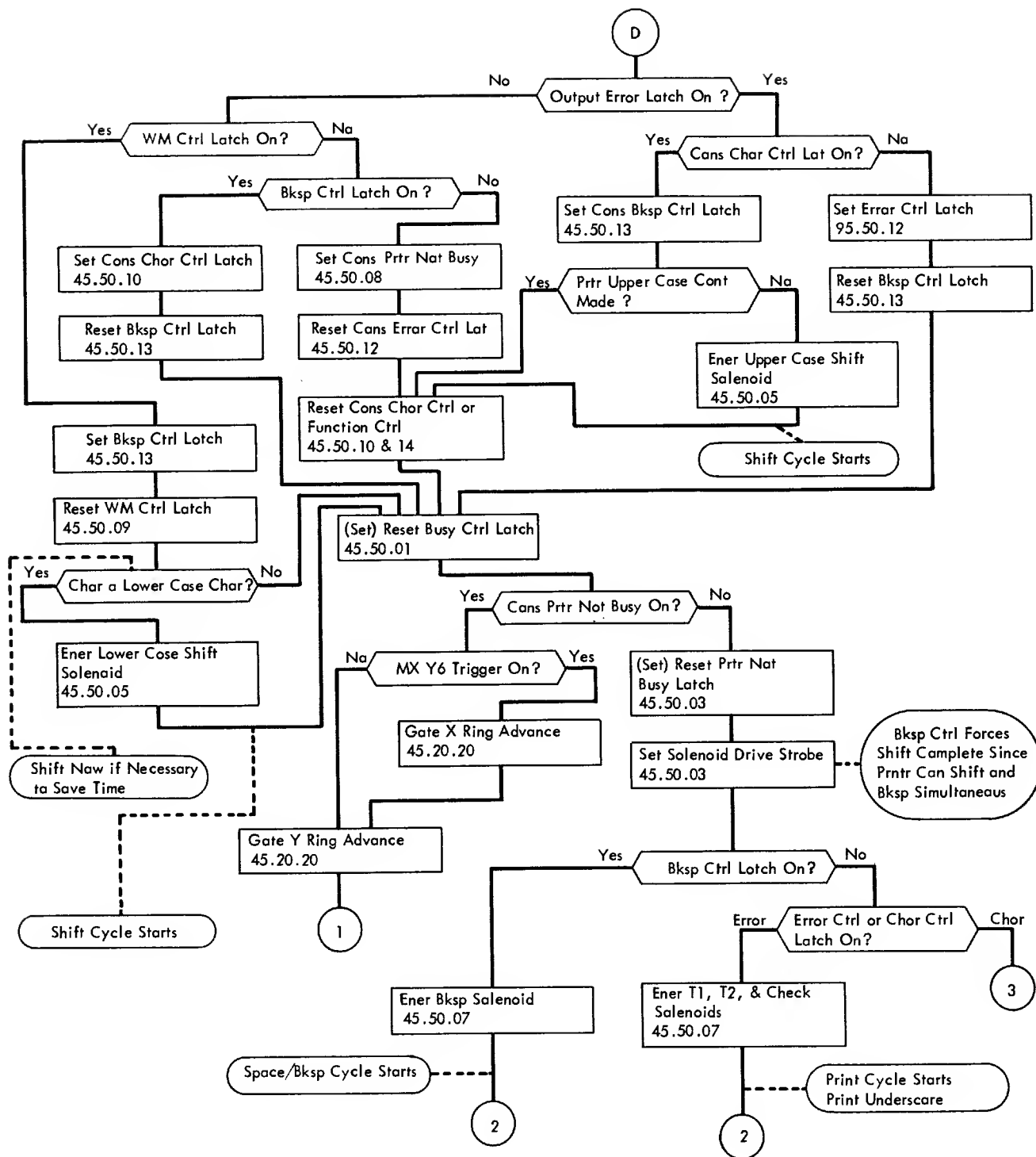


Figure 7.1-5C Printer Operation (Output)

error latch is still ON but the Cons Char Ctrl is not ON. This condition sets the Error Ctrl latch to reset the Bksp Ctrl latch, and gates the T1, T2 and check solenoids to print the underscore. The solenoid driver strobe again controls energizing the solenoids.

When the check strobe sets, the Error Ctrl latch gates the reset of the output error latch. At the end of the print cycle, the Cons Prtr-Not-Busy latch sets and the matrix advances to the next position.

#### 7.1.07 Word-Mark Control (Figure 7.1-5)

##### Output

When the matrix gates a character to the console output lines, the presence of a word mark sets the console word-mark control latch. This latch causes a word mark to print. This is followed by a backspace and another print cycle to print the character. The console character-control latch, is not set for the first print cycle. This prevents the output lines from energizing the rotate and tilt solenoids. Setting the word-mark control latch resets the console printer-not-busy latch that prevents the matrix ring from advancing. If the printer is not already in upper-case shift, the WM control latch causes a shift cycle. When the console printer-shift-complete line comes up, the solenoid driver-strobe latch sets and the word mark (inverted circumflex) is printed. The control printer-strobe trigger is not turned ON during the word-mark print cycle because the console-character control latch was not turned ON.

When the check strobe is reset at the end of the word-mark print cycle, the backspace control latch is turned on. Setting the backspace control latch allows a shift cycle to occur if the character to be printed under the word mark is a lower-case character. The backspace control latch also keeps the console printer shift-complete-line up to set the solenoid-driver strobe.

The solenoid driver-strobe energizes the backspace solenoid since a shift and backspace cycle can be done simultaneously. When check strobe is reset at the end of the backspace cycle, the console cycle-latch-set pulse sets the console character control latch. The solenoid driver strobe energizes the rotate and tilt solenoids depending on the character that is to be printed. The character is printed under the word-mark symbol in the same manner as in a normal print cycle. At the completion of the print cycle, the console-printer-not-busy latch sets. This allows the matrix ring to advance.

##### Input

When it is desired to enter a WM into the CPU, first press the WM key to a Space/Bksp/WM cycle. During the cycle the WM N/O contact transfers to set the WM Ctrl latch, if the operation is a console write or alter operation.

The WM Ctrl latch locks the keyboard and starts the same WM routine as in output mode. During the backspace cycle Alter Inquiry Unlock, and Bksp Ctrl latch energize the LC shift solenoid to return the printer from UC. At the end of the Bksp cycle the keyboard again unlocks and the operator proceeds with the operation.

#### 7.1.08 Carriage Return (Figure 7.1-5)

Console matrix positions 6A and 34 or console printer end-of-line start a carriage-return operation by bringing up the console carriage-return line. The console printer-not-busy latch resets to prevent the matrix from advancing during the carriage-return operation. Energizing the carriage-return solenoid starts the carriage-return cycle. The carriage-return interlock contact transfers at 145° and remains transferred until the carriage fully returns. When the contact transfers at 145°, the console check-strobe trigger sets and remains set until the contact transfers back to normal when the carriage fully returns. Resetting the console check-strobe trigger allows the console cycle-latch-set and the console cycle-latch reset lines to be brought up. The console cycle-latch-set line sets the console printer-not-busy latch. This allows the matrix to advance to the next or home position, depending on the console operation. The console cycle-latch-reset line resets the function control latch and ends the operation.

#### 7.1.09 Stop Print-Out (Figure 7.1-6)

A stop print-out operation is started by any one of three major conditions. The three major conditions are: normal stop, single-cycle stop, and error stop.

A normal stop occurs with any one of these conditions:

1. A program stop Op code.
2. A pressed stop key.
3. Rotating of the mode switch to another position.
4. An input-output error when the machine is in the 1401 mode and the input-output-check stop-switch is ON.
5. The address stop switch on 1411 CE panel is turned ON.

A single cycle stop occurs at the completion of the instruction read-out and at the end of an execute cycle when in the instruction-execute mode, at the completion of each memory cycle when in the storage cycle mode, and at the end of each logic gate pulse when in the logic step mode.

An error stop occurs whenever the master error line energizes.

A stop print-out operation cannot occur until the central processing unit clock stops. The CPU clock cannot stop, if the E- and F-channels are in use. After the CPU clock stops, the console enable-stop print-out line comes up to start a carriage return. If the stop condition latch is OFF when a stop print-out operation is signalled, the 1410 system stops and no printing takes place.

The stop condition latch is ON, except when the mode switch is in the display or alter positions, or except when the print-out inhibit switch is ON. At the completion of the carriage-return cycle, the console stop-carriage-return complete line is brought up to allow the console stop-print latch to set. The console stop-print latch gates the console matrix into position 35, where an S is printed for a



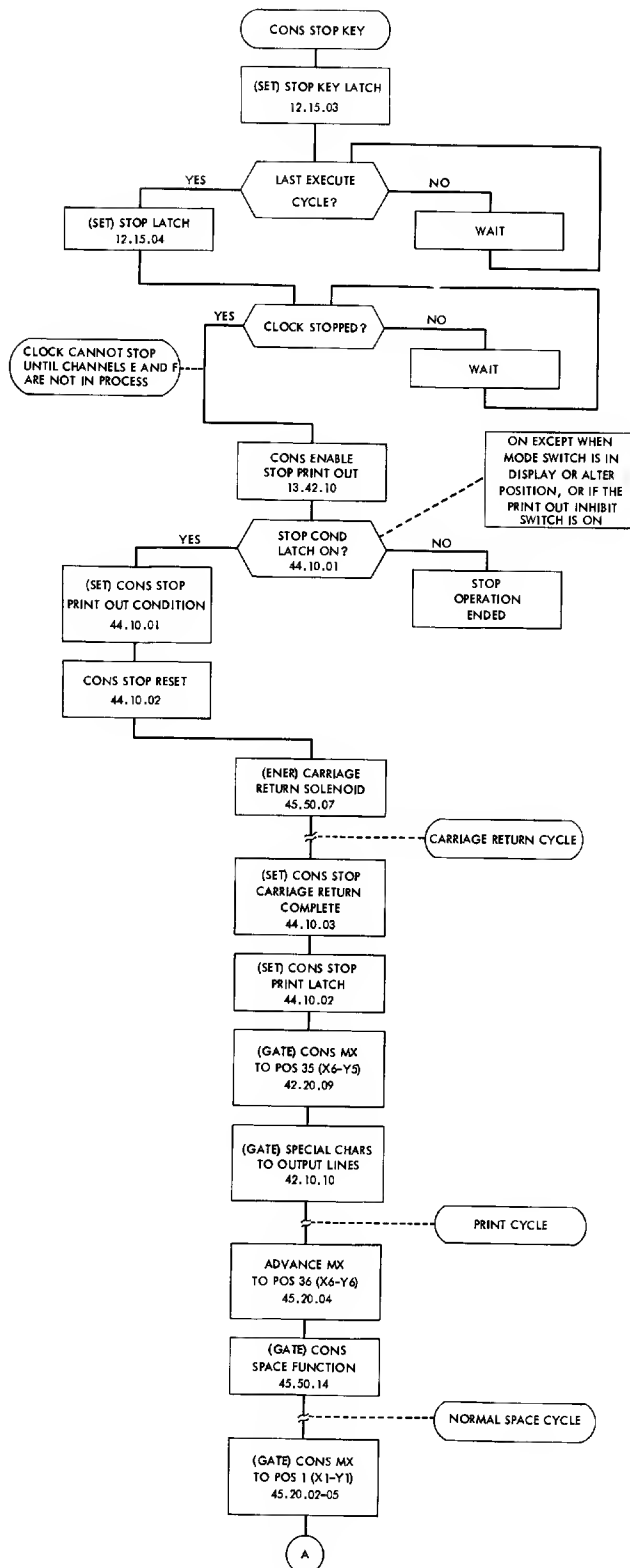


Figure 7.1-6A Stop Print-Out Operation

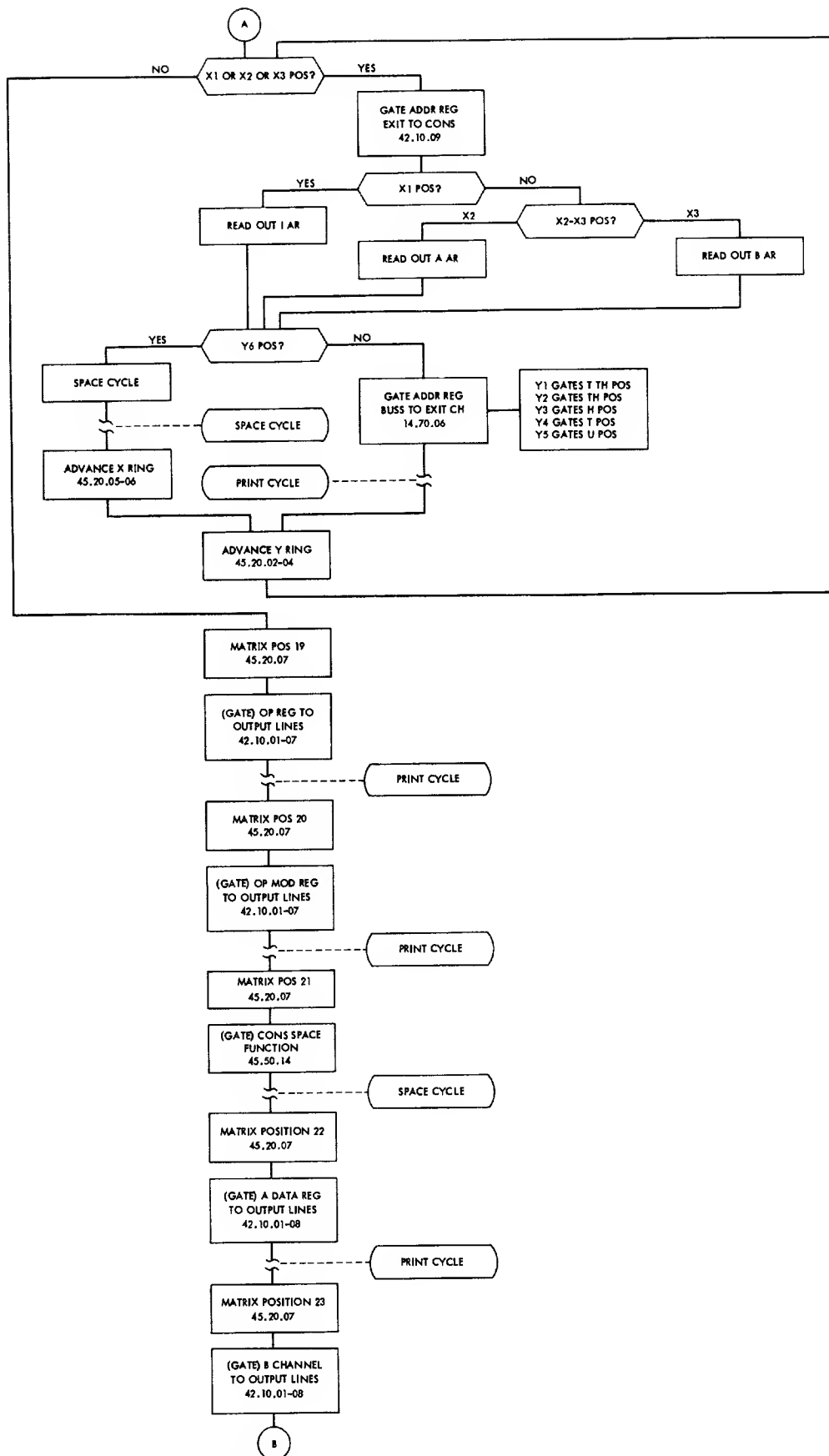


Figure 7.1-6B Stop Print-Out Operation

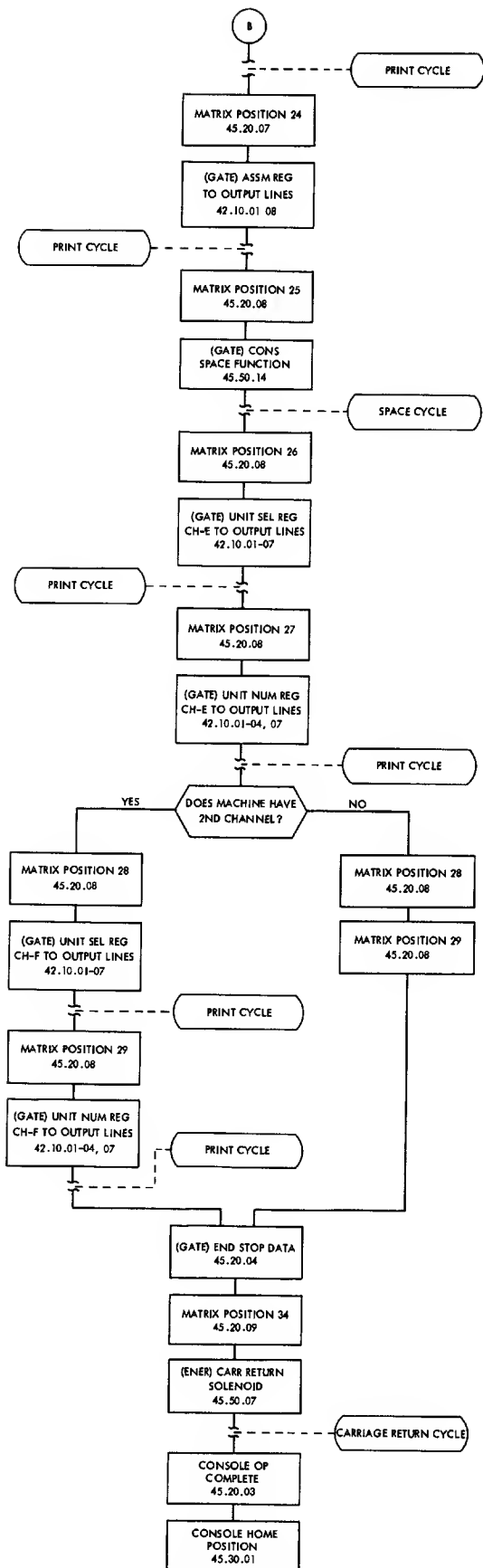


Figure 7,1-6C Stop Print-Out Operation

normal stop, a C for a single cycle stop, and an E for an error stop. After the special character is printed, a space occurs in matrix position 36, and the matrix advances to position 1 to begin the print-out of the IAR register.

Whenever the X1, X2, or X3 triggers in the control matrix are ON, the console address-register exit-gate line gates the address-register exit-channel to the console. The console matrix X1-position line gates the IAR register to read out to the address bus. Console matrix Y1-position gates the ten-thousands-position digit from the address bus to the printer. Console matrix Y2-position prints the thousands-position-digit, Y3 the hundreds-position, Y4 the tens-position, and Y5 the units-position. Console matrix Y6 combines with X1 or X2 or X3 (positions 6, 12, and 18) to start a space cycle. In a similar manner, the AAR and the BAR registers print out with console matrix X2 and X3 positions respectively.

Matrix position 19 gates the Op code register to the console output lines. Figure 7.1-2 illustrates the unit of the machine that is gated to the console output lines for each console matrix position. Matrix positions 28 and 29 gate the unit-select register and unit-number register (channel F) to the console output lines if the machine has two I/O channels. If the machine has only one I/O channel, the matrix advances through positions 28 and 29 without printing. Take Console Printer Cycle is blocked by positions 28 and 29 to prevent setting the reset-printer-not-busy latch. Cons Prtr-Not-Busy latch stays ON to gate Cons Clock Two Position pulses to advance the matrix through position 29.

The matrix skips from position 29 to 34 and a carriage return starts in position 34. After the carriage return, the matrix returns to home position and the console stop print latch resets.

With both the console-stop print-latch and the stop-condition latch OFF, the console-stop-print-complete line comes up. This allows the operator to start another operation.

#### 7.1.10 Inquiry

The 1415 console can be an inquiry station by use of the console inquiry keys. The keys are operated in the following order during an inquiry request operation.

##### Request Key

Start an inquiry by pressing the request key. The request key turns the console inquiry request latch ON. The programmer can arrange the program to operate in the following manner:

1. The program advances to a condition test instruction J(I)Q.
2. If the console inquiry request latch is ON, the program branches to a read console printer instruction, M/L%T0(B)R.
3. If the inquiry request latch is OFF, the program continues until the J(I)Q instruction is executed again.

4. When the inquiry request is granted, the console printer instruction unlocks the keyboard so the operator can enter keyed data in the storage area designated by the B-address.

#### Release Key

After the correct number of characters enter, press the release key. The programmer has already indicated the correct length of the B-field of the M(%T0)(B)R instruction by placing a group-mark with a word-mark in the next higher storage location. After the last inquiry character enters in storage, press the release key. Another character reads out of storage as a result. This character must be a GM·WM in order to obtain a correct-length record check.

Pressing the release key also returns the carriage, locks the keyboard, and resets the inquiry request latch.

#### Cancel Key

The inquiry routine in process can be ended by the cancel key in the same way as with the release key. Pressing the cancel key causes the same functional operations as the release key, except that the E-channel-condition status indicator sets. The programmer must then test the condition indicator to determine that a cancel took place. A carriage-return and an index operation occur after cancel, and the program resumes.

#### Read Console Printer, M%T0(B)R or L%T0(B)R

After pressing the inquiry request key and the console inquiry request latch sets, the program advances to a condition-test branch instruction that tests the inquiry request latch. If it is ON, the program branches to a read-console-printer instruction.

#### Op-Code Function

The following sequence of events takes place after the read-console-printer command is detected:

1. The console printer prints the letter I.
2. A single space occurs.
3. The keyboard unlocks.
4. The operator enters characters from the keyboard. The first inquiry character is placed in the B-field address specified by the read console command. Subsequent characters are placed in next higher storage positions.
5. After the last character enters, press the release key. If the group-mark with a word-mark that was placed in storage by the programmer reads out at this time, correct-length record is signalled and the program advances to the next instruction.

6. A carriage return starts after each printing line and after pressing the release key. The keyboard locks at the end of the operation.

#### Operation (Figure 7.1-7)

When either an M or L%T0(B)R instruction is given, the select unit T-line comes up to select the console. The E-channel status indicators reset. If a read-console-printer instruction is given without the console-inquiry-request latch ON, the E-channel status sample A-pulse sets the no-transfer status latch. A last execute cycle is signalled and the program moves to the next instruction.

If the test of the status latches indicates that none are ON, the console read Op line turns ON. If the operation is M, the console move-read Op line also comes ON. The following sequence of events occurs:

1. Print I. The console read Op line causes the matrix to set to position 30. This forces the printing of the letter I to identify the operation as an inquiry.
2. Space. The matrix steps into position 31 where a printer space operation occurs.
3. Unlock keyboard. After the space, the matrix steps into position 32. This unlocks the keyboard so the operator can key the inquiry request.
4. Gate Character to E1. The console read Op line sets up a gate to switch the console output to the input gate on the E1 register. When the operator presses a character key, a console strobe pulse results. This sets the BCD-coded character in the E1 register.
5. Enter word marks. If an L%T0(B)R command is being executed, the operator may enter word marks as in an alter operation. If an M%T0(B)R instruction is given, no word marks can be entered by pressing the word-mark key.
6. Transfer character to E2. When E2 is not full and E1 is full, a set-E2-signal sets the E2 register from E1.
7. Store E2 in B-field. An E-cycle now takes place. This stores the contents of E2 in the B-address. The character is gated from E2 to the A-channel, and to assembly. The BAR is modified by plus-one during the store cycle.
8. Inquiry release. In entering the complete record, press the inquiry release key. This generates an external end-of-transfer signal. Another E-cycle is signalled, and the next character that reads out should be a group-mark with a word-mark. If it is, the end-of-record latch sets and the B-channel zone, numerical, and WM are stored to keep the group-mark with a word-mark intact in storage.

A test is now made for a wrong-length record. Because no character key was pressed to read out the GM•WM, the E2 full latch and the E-channel strobe trigger should be OFF. If either are ON, a wrong-length record is signalled. Otherwise, correct-length record sets. In either case, the channel-in-process latch resets and last execute cycle sets (Unovlp). The wrong-length-record status indicator must be tested by the programmer after the program advances.

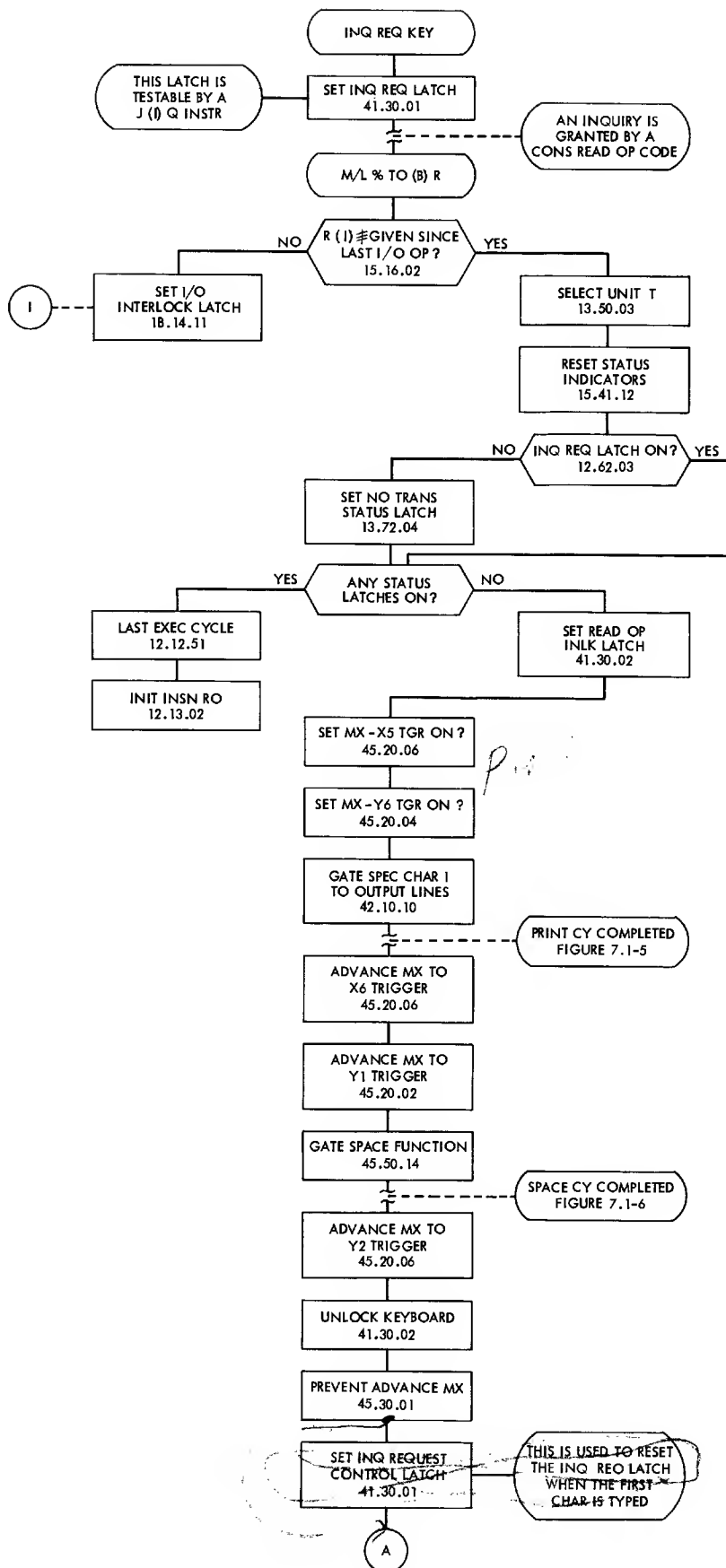


Figure 7.1-7A Inquiry Request and I/O Printer Read





Figure 7.1-7C Inquiry Request and I/O Printer Read

If the operator presses the release key before entering the last character of the message, E1 and E2 Not Full, set the last input cycle. With last input cycle ON and no GM·WM, the internal end-of-transfer latch sets but end-of-record remains OFF. The test for correct-length record now finds both E2 full and E-channel strobe OFF. But end-of-record is also off, so that wrong-length record sets.

If a character key operates after the correct number of characters is entered, a GM·WM is sensed. The character is not stored, and the end-of-record latch sets. Because no external end-of-transfer is generated, nothing happens until the operator presses either the cancel or release key. External end-of-transfer is set, and with internal end-of-transfer ON. The wrong-length record can be set because E2 Full is ON.

9. Inquiry Cancel. The cancel key operates the same as the release key except that the condition latch is set. This is tested by the programmer to determine that a Cancel takes place. If the cancel key is operated after the correct number of characters enter, the correct-length record sets. If the cancel key is operated before the correct number of characters enter, the wrong-length record turns on and the indicator must be tested by the programmer.
10. Lock keyboard and single carriage return. When E-channel status sample B comes on after either the release key or cancel key is pressed, E-Ch-in-Process turns off. This causes the reset of the console read Op signal (41.30.02). With console read Op OFF, the inquiry keyboard unlock line drops and the keyboard locks.

The Ptr Locked Cnd Proceed line comes back on when the keyboard locks. This allows the matrix drive to continue through position 33 to 34, where a carriage return is signalled. The matrix then advances to the home position.

#### 7.1.11 Write Console Printer, M%T0(B)W or L%T0(B)W

##### Op-Code Function

The console write command causes the following sequence of events.

1. The character R types (signifies a reply).
2. A single space occurs.
3. The data from the B-field transfers from storage and is printed by the console I/O printer until a group-mark with a word-mark is sensed in storage. The I/O printer spaces when a valid blank is sensed in storage on a M-Op. On a L-Op, a b prints for a blank. Inverted circumflexes print for word marks on L-Ops only.
4. Carriage return and index operations occur at the end of each printer line, and at the end of the operation.
5. The program continues with the next instruction (Unovlp).

#### Operation (Figure 7.1-8)

When the M or L instruction reads out, the E-channel reset line resets all the I/O status latches. If the console cycle-control matrix is not in home position, the E-channel status sample A-pulse sets the E-channel busy status indicator. This situation occurs if at the end of a previous-read console-printer operation, the carriage return was not complete. The console matrix will not be returned to the home position. The status indicators are tested, and if any are on, a last-execute cycle (unoverlap mode) is signalled, and the program goes on to the next instruction.

If the status latches are OFF, the console write Op signal develops. This causes the matrix to set at position 30. Matrix position 30 prints the special character R to identify the print-out. The matrix advances to position 31 where a space operation starts. At the completion of the space, the matrix moves to position 32.

While the matrix-position-30 operation takes place, the first E-cycle is completed to fill E1. The character in E1 transfers to E2 and a second E-cycle takes place to fill E1. Thus, by the time the matrix advances to position 32, both E1 and E2 are filled. Console write Op switches with console matrix position 32 to gate the character in E2 to the console output lines. The character prints and a console strobe pulse is sent back to the CPU to cause the reset of the E2 full latch. With the E2 full latch OFF, the E2 register is filled from E1, and another E-cycle (Unovlp) is taken to refill E1. With E2 filled, another character can be printed.

The following sequence of events takes place to end the operation.

1. The E2 full latch resets after the next-to-last character prints.
2. The last character transfers from E1 to E2.
3. Not Int End-of-Transfer causes another E-cycle in which the group-mark with a word-mark reads out of storage. This sets E-Ch Int End-of-Transfer and prevents the setting of E1 with the B-channel character.
4. With E2 full, another print operation takes place to print the last character. E2 Full resets OFF as usual.
5. With both E1 Full and E2 Full OFF, Ext End-of-Transfer sets. This turns on E-Ch Status Sample B to force the last execute cycle (Unovlp).
6. The E-channel status sample B-delay impulse resets E-Ch Unovlp-in-Process and, in turn, E-Ch-in-Process. When E-Ch-in-Process drops, Console Write Op goes OFF.
7. With Console Write Op OFF, the matrix advances to position 34 where a carriage return is set up.

Any errors that occur during the operation, except an I/O printer error during transfer, stop the system.

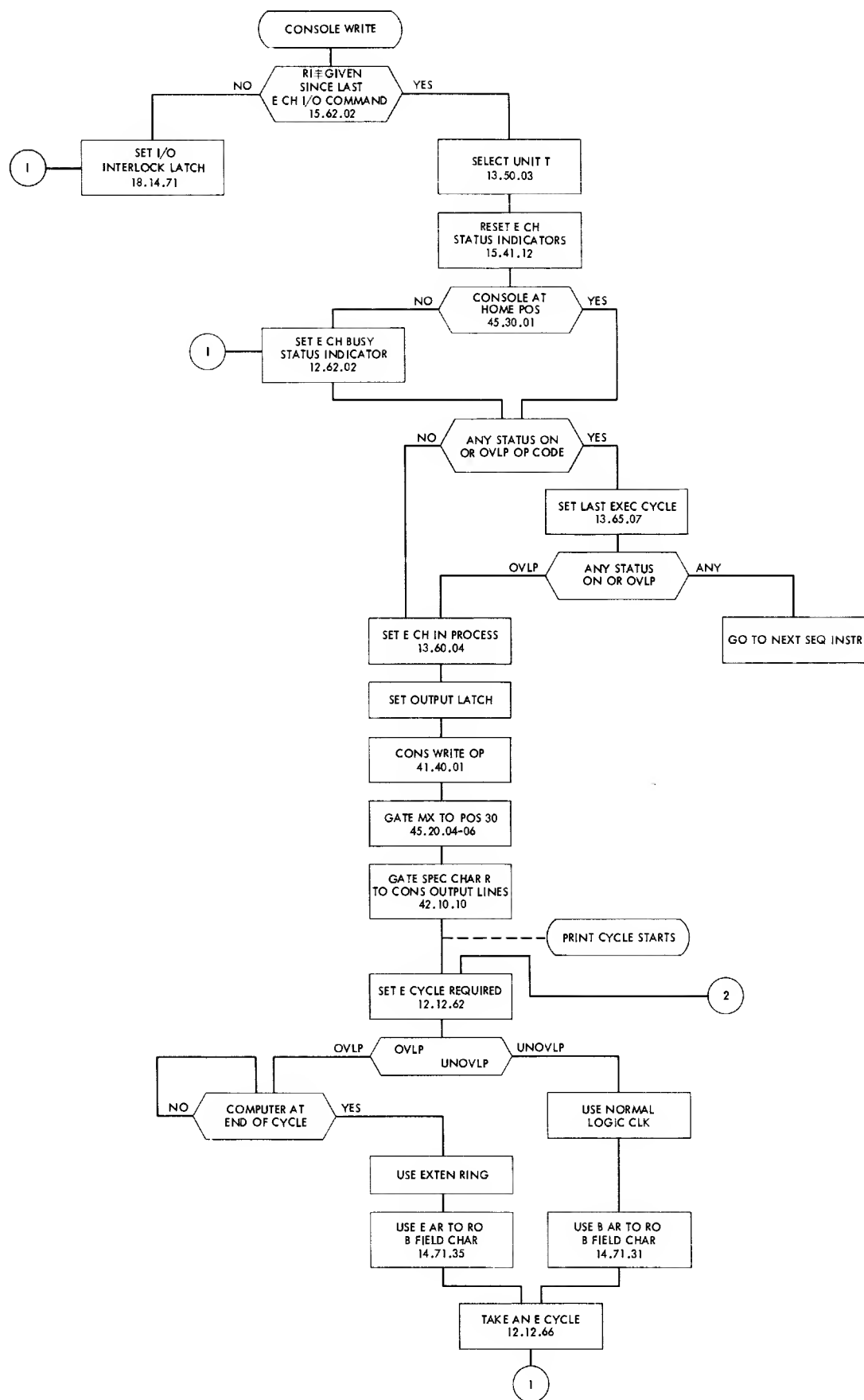


Figure 7.1-8A Console Write

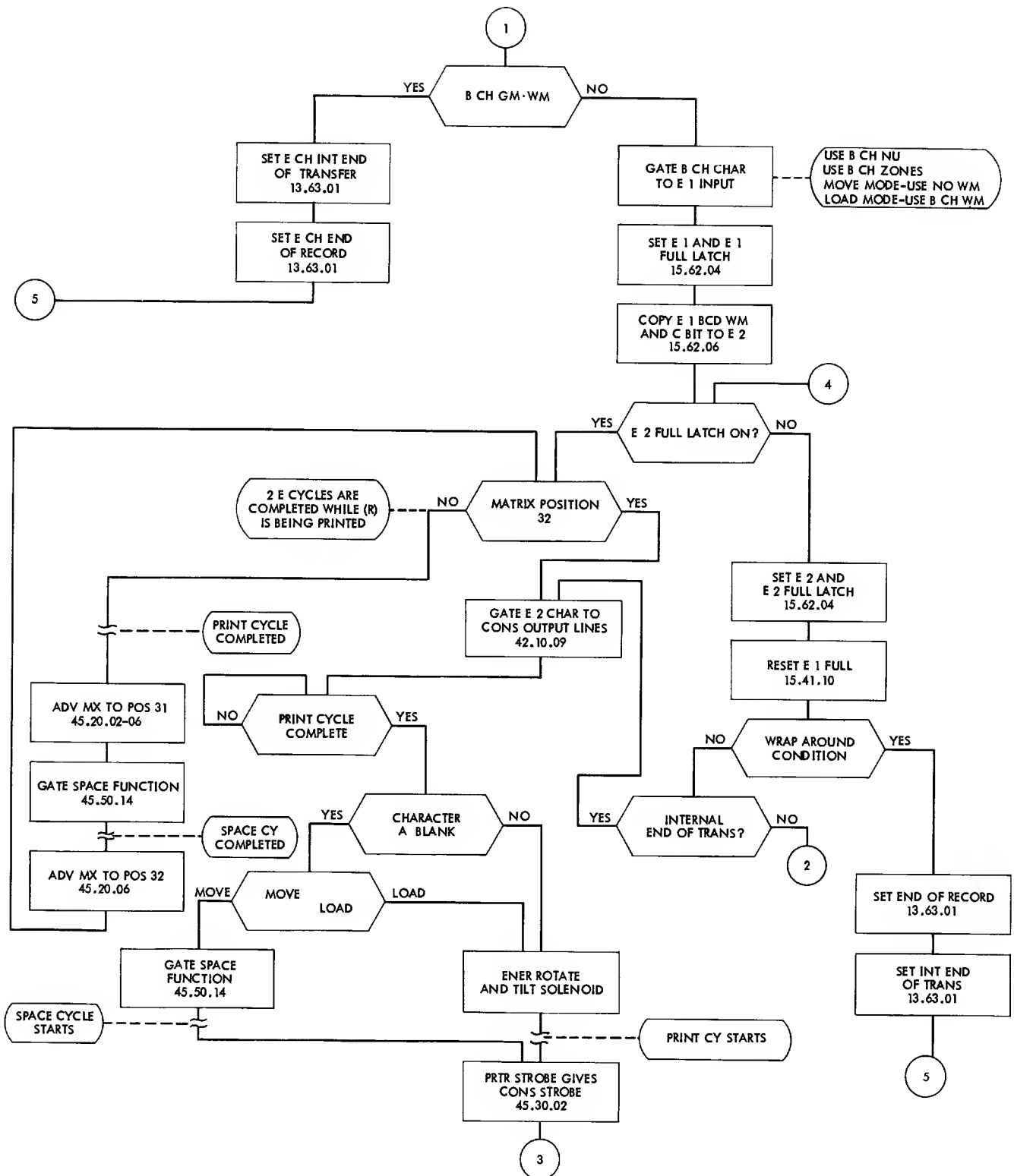


Figure 7.1-8B Console Write

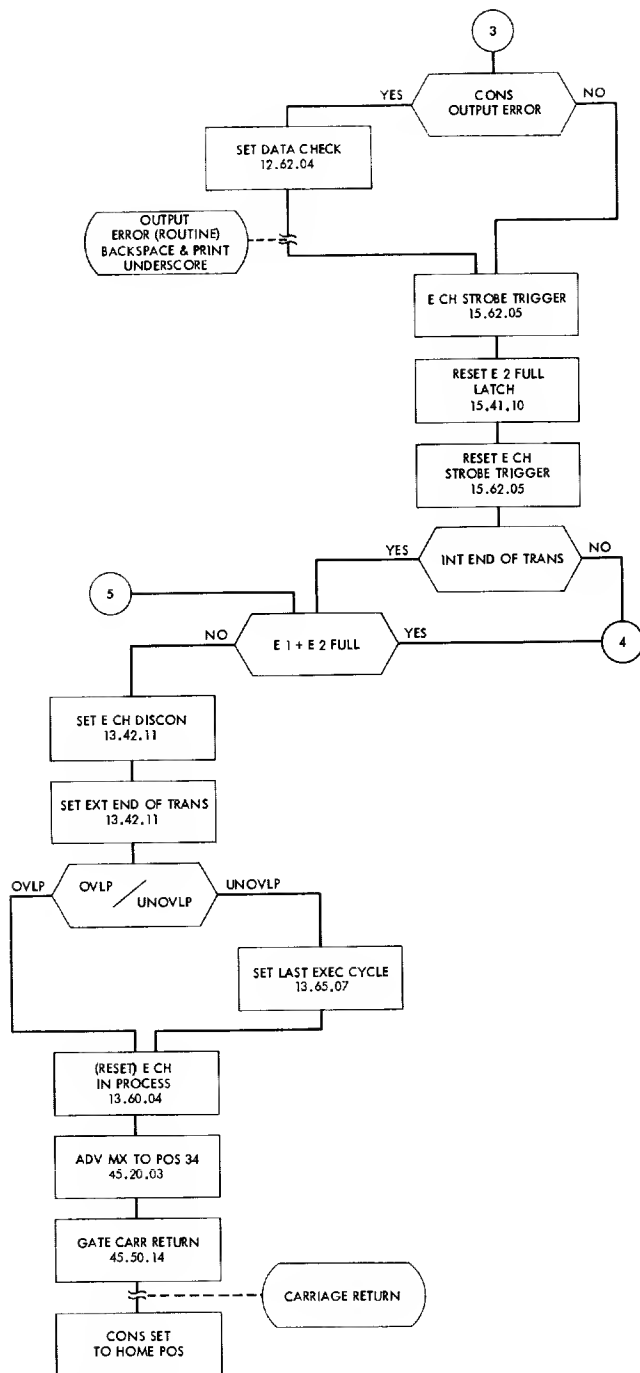


Figure 7.1-8C Console Write

### 7.1.12 Console Address Set

#### Description (Figure 7.1-9)

The console address set feature of the 1415 permits an address to enter into any address register in the CPU by means of the input-output printer. The operation starts by rotating the mode switch to the address set position. If the system has not already stopped, the rotation of the mode switch from one position to another stops it, and initiates a stop print-out operation. The address register that is to be altered is selected with the address entry switch on the console CE panel. If this switch is set to Normal, the address set by the operator enters IAR. When the system starts in one of the run modes, the program starts at the new address. Pressing the start key starts the address-set routine, if the CPU clock was stopped at logic gate A. The system always stops at logic gate A except when the previous operation was in the single logic step mode. The start key must be pressed after the completion of stop print-out. Pressing the start key resets the address register that the address entry switch selected.

If the address entry switch is in the normal position, all branch latches reset, except the no-branch latch, which sets. This insures that the program restarts with the address set in IAR.

The necessary gates are now brought up to reset the register that is selected by the address entry switch, and to gate the console matrix to position 35. In matrix position 35, the special character that identifies the operation prints. If the address entry switch is in the normal position, a B prints. If the switch is in any other position, a number sign (#) prints. When the console printer-not-busy latch sets at the completion of the special character print cycle, the matrix advances to position 36, and initiates a space cycle.

When the space operation is complete, the matrix advances to position 1A (X1A-Y1). At this time, the keyboard unlocks and the necessary gates are brought up to allow information from the console to enter the address register that the address entry switch selects. When the key on the printer is pressed, the first character prints. When the latch contacts make, they are decoded into BCD code and gated to the tens-thousands position of the selected address register.

At the completion of the print cycle, the matrix advances to the Y2 position. The second character enters the thousands position of the register. This operation continues until the fifth character enters the units position when Y5 is ON. When the last digit of the address enters in the register, the address-set routine latch resets, the keyboard locks, and the matrix advances to the Y6 position. Once the keyboard is locked, a carriage return operation starts. When the carriage return operation completes, the matrix advances to home position (X6 trigger ON, all Y triggers OFF).

There is no error detection during the address set operation, because the CPU stops and none of the error circuits are sampled.

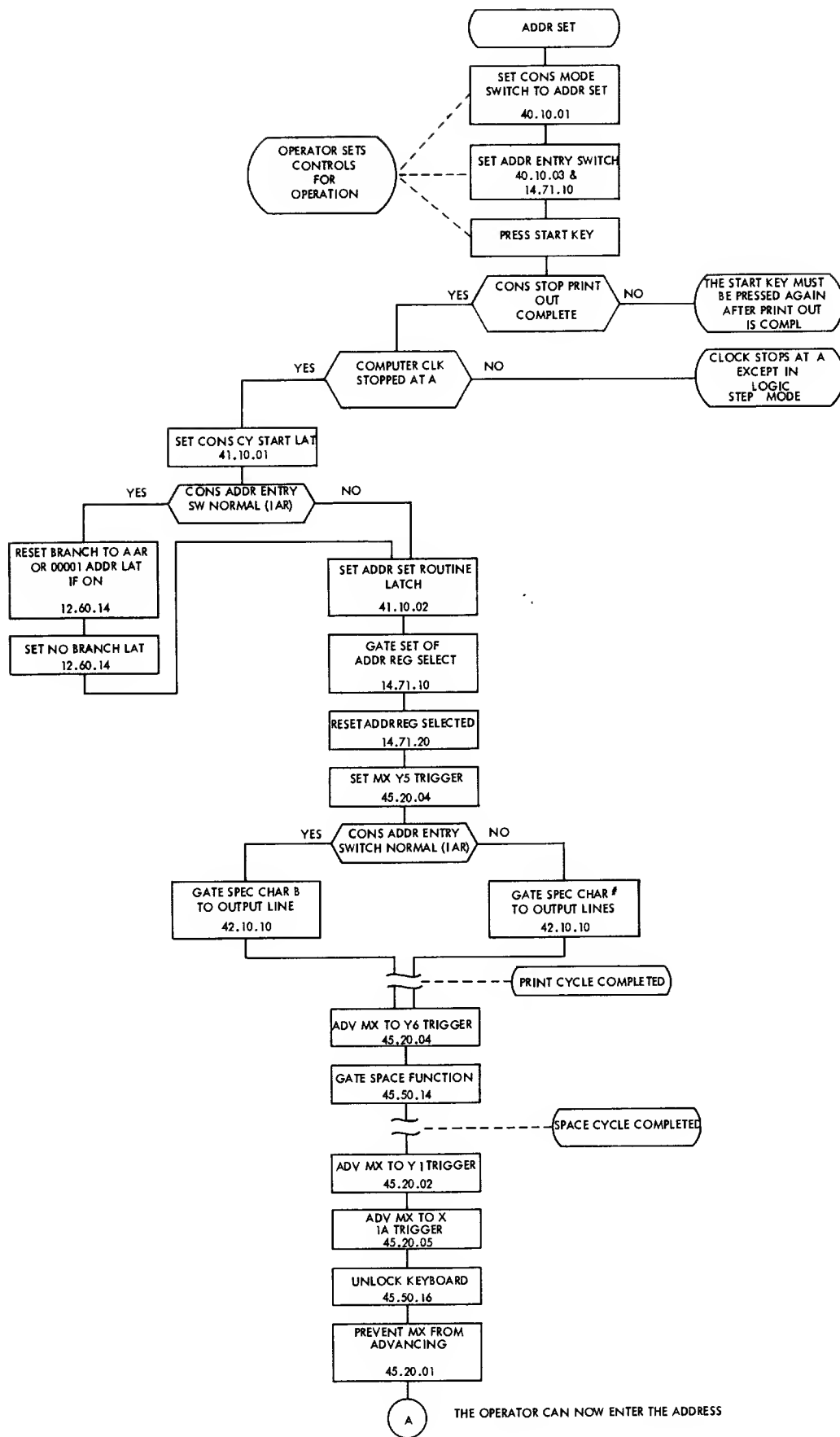


Figure 7.1-9A Address Set Routine



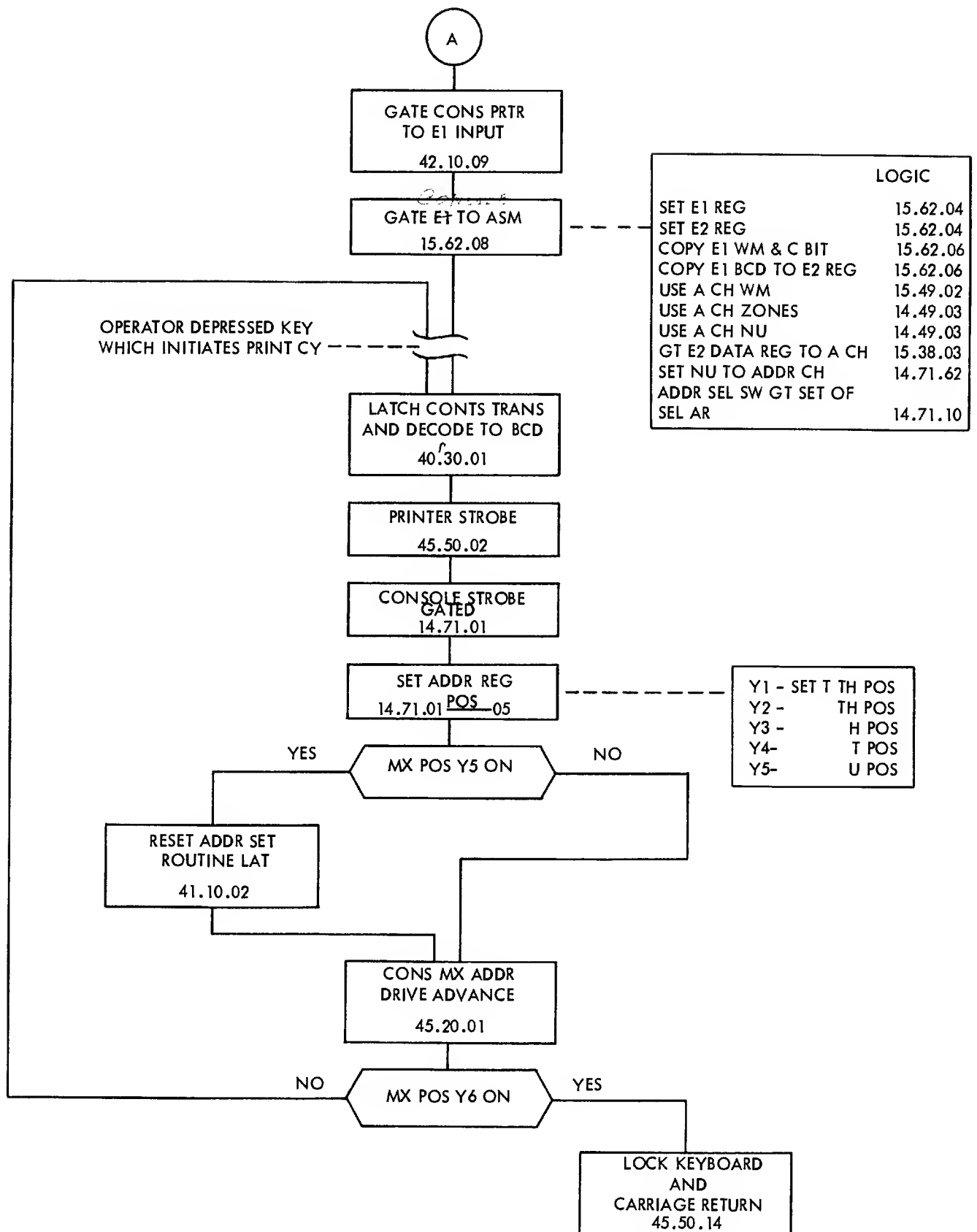


Figure 7.1-9B Address Set Routine

### 7.1.13 Console Display

#### Description (Figure 7.1-10)

The display routine allows the operator to print the contents of any location in storage on the console input-output printer. The display can be as short as two characters or it can be many lines in length.

To initiate a display operation, the mode switch is rotated to the display position. If the system did not stop before moving the mode switch, it stops when the switch moves, and a normal stop print-out follows. With the console control matrix returned to its home position following the stop print-out, and the CPU stopped at the last execute cycle, pressing the start key starts the display operation. The display operation starts with an address set routine during which the address of the area in storage that is to be displayed reads into the CAR.

The CAR resets and the console matrix sets to position 35. The special character D prints at matrix position 35 to identify this as a display operation. When the console printer-not-busy latch sets at the completion of the printing of the D, the matrix advances to position 36, where a space operation starts. At the completion of the space cycle, the matrix advances to position 1A (X1A-Y1) and the keyboard unlocks. Key in at this time the five digits of the address for the storage location of the first display character. The five digit address reads into the CAR in the same way as described in the section, 7.1.12 Console Address Set.

After the address sets in the C-address register, and the carriage-return cycle completes, the matrix skips from position 6A to position 30 where another D prints. While the console printer is printing the D, a console strobe pulse is sent to the CPU. The console strobe pulse starts the CPU clock and the CPU takes a D-cycle. During the D-cycle the address in CAR reads into the storage address register. It is modified by plus one, and stored in the D-address register. Storage-selection circuits set up to read out the character from storage into the B-register and B-channel.

At the completion of the D-print cycle, the matrix advances to position 31 where a space cycle starts. At the end of the space cycle, the matrix advances to position 32. Matrix position 32 gates the B-channel to the console output lines to permit the first character to print. During the print cycle, the console strobe pulse is sent to the CPU. The CPU allows the modified address in the D-address register to read into the storage address register, to be modified by plus one, and to be stored back in the DAR.

Storage-selection circuitry allows the next character in storage to read out and to enter the B-register and B-channel. The word mark associated with the first character does not stop the operation when the matrix is in position 32. Because this is a reverse scan operation, the word mark that is associated with the high-order character in the field will stop the operation, if it is not disregarded. At the completion of the first-character print cycle, the matrix advances to position 33. Matrix position 33 gates the B-channel to the console output lines, and the second character prints.

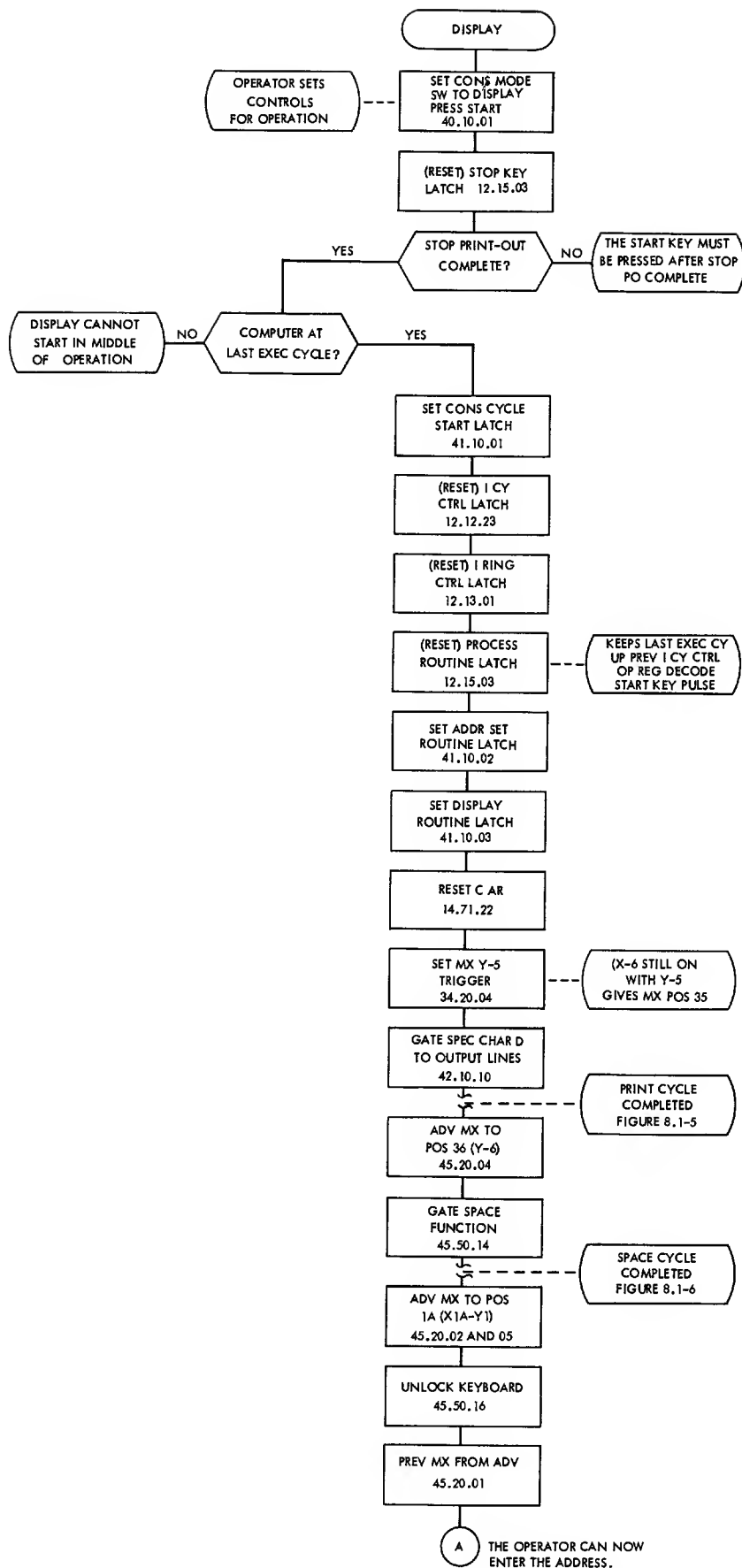


Figure 7.1-10A Display Operation

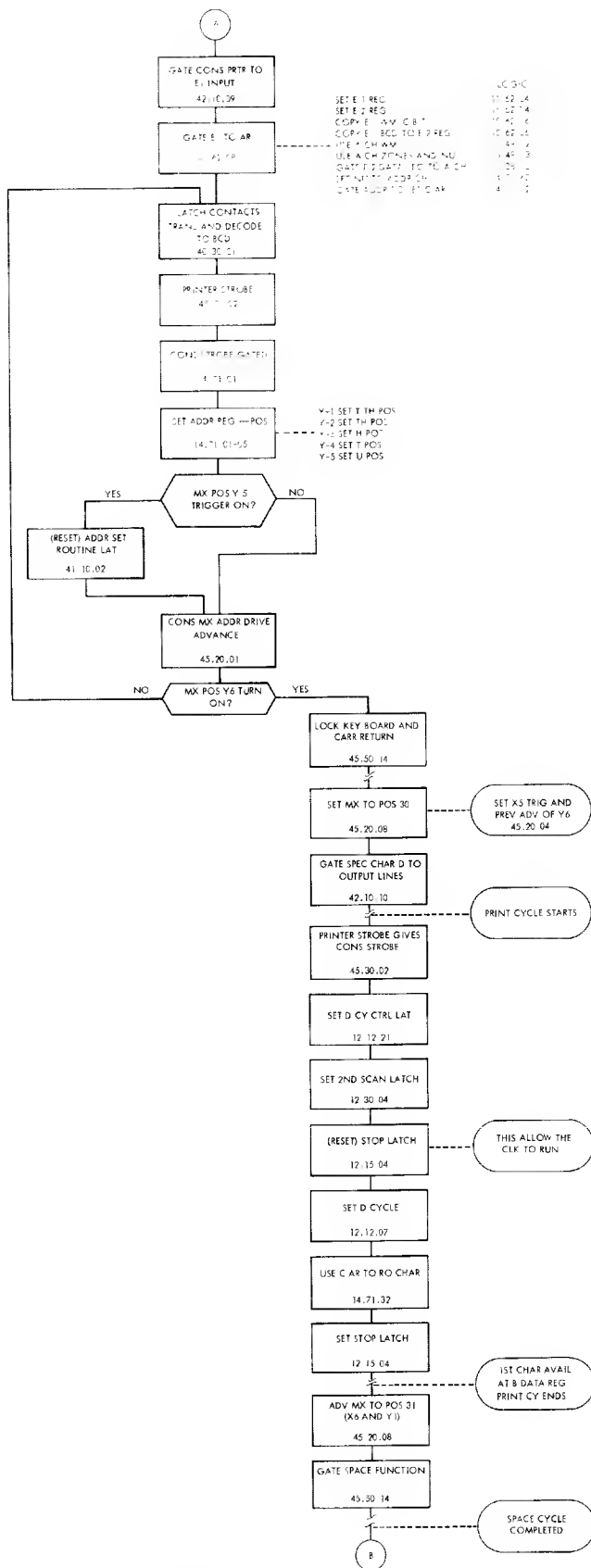


Figure 7.1-10B Display Operation

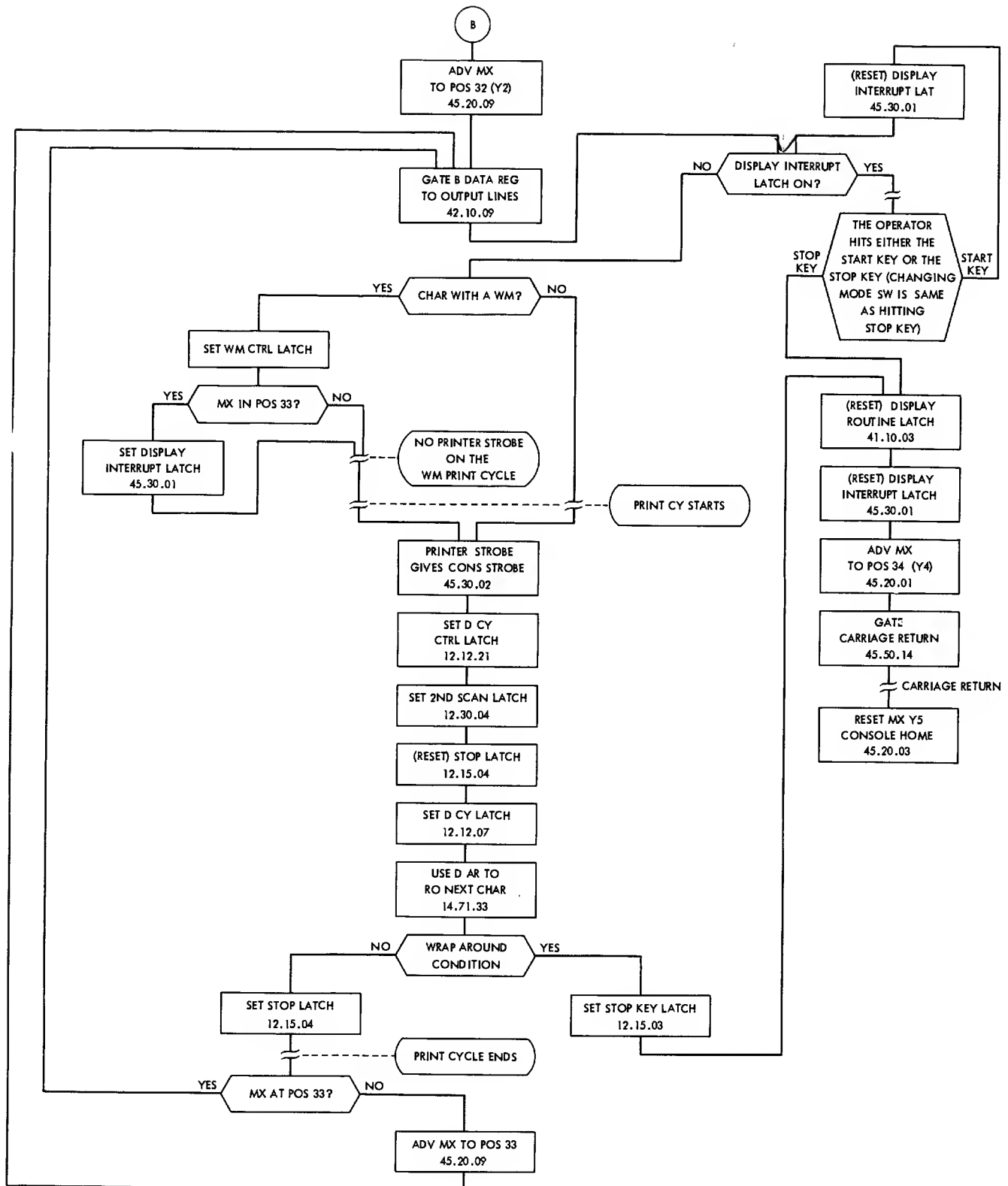


Figure 7.1-10C Display Operation

During the print cycle, the console strobe pulse again causes a D-cycle in the CPU. The address in DAR reads into the storage address register, is modified by plus one, and is stored back in the DAR. Storage-selection circuitry allows the third character in the field to enter the B-register and B-channel. The remaining characters in the field read out and print in a similar manner.

Characters print in position 33 until a B-channel word mark is sensed. Each character sends a console strobe pulse to the CPU. This causes a D-cycle to occur as previously described. When the display stops after the B-channel word mark, one full field has printed plus the first character in the second field with its word mark. The second character of the second field is left on the B-channel.

If the operator now wishes to display another field, pressing the start key allows the display to continue until another B-channel word mark is encountered. If it is not desirable to have the display stop after every word mark, the start key can be held down. If the end of a line is reached in a display operation, the carriage returns to the left-hand margin, and the display continues on the next line.

The display can be discontinued at any time either by pressing the stop key or by rotating the mode switch out of the display position. This action resets the display routine latch at the completion of the print-out of the character that it is currently handling. Resetting the display routine latch advances the matrix to position 34 where a carriage return starts. Upon completion of the carriage return, the matrix advances to home position.

Data errors are ignored on display. Other errors halt the display and must be cleared by resetting or pressing the start key in one of the run modes. No error print-out occurs.

#### 7.1.14 Alter Routine

Description (Figure 7.1-11)

An alter routine permits the operator to change the data in any location of memory, if this data has been previously displayed.

The display routine that must be executed before the alter routine is the same operation as described in the section 7.1.13 Console Display. The operator can alter one field or one full line depending on what has previously been displayed.

When the display message is printing out during a display operation, the first B-channel word mark that is encountered (after the first character of the display) sets the word mark condition alter latch. If the display message extends long enough to cause the end-of-line latch to set (one full line), the word mark condition alter latch resets and the full-line-condition alter latch sets. These latches set during the display and remain set until the alter operation ends or an operation other than the alter starts.

The alter operation starts by rotating the mode switch to the alter position. Press the start key to gate the console control matrix into position 30, where special character A prints to identify the operation as an alter routine. At the completion of the A-print cycle the matrix advances to position 31. In position 31

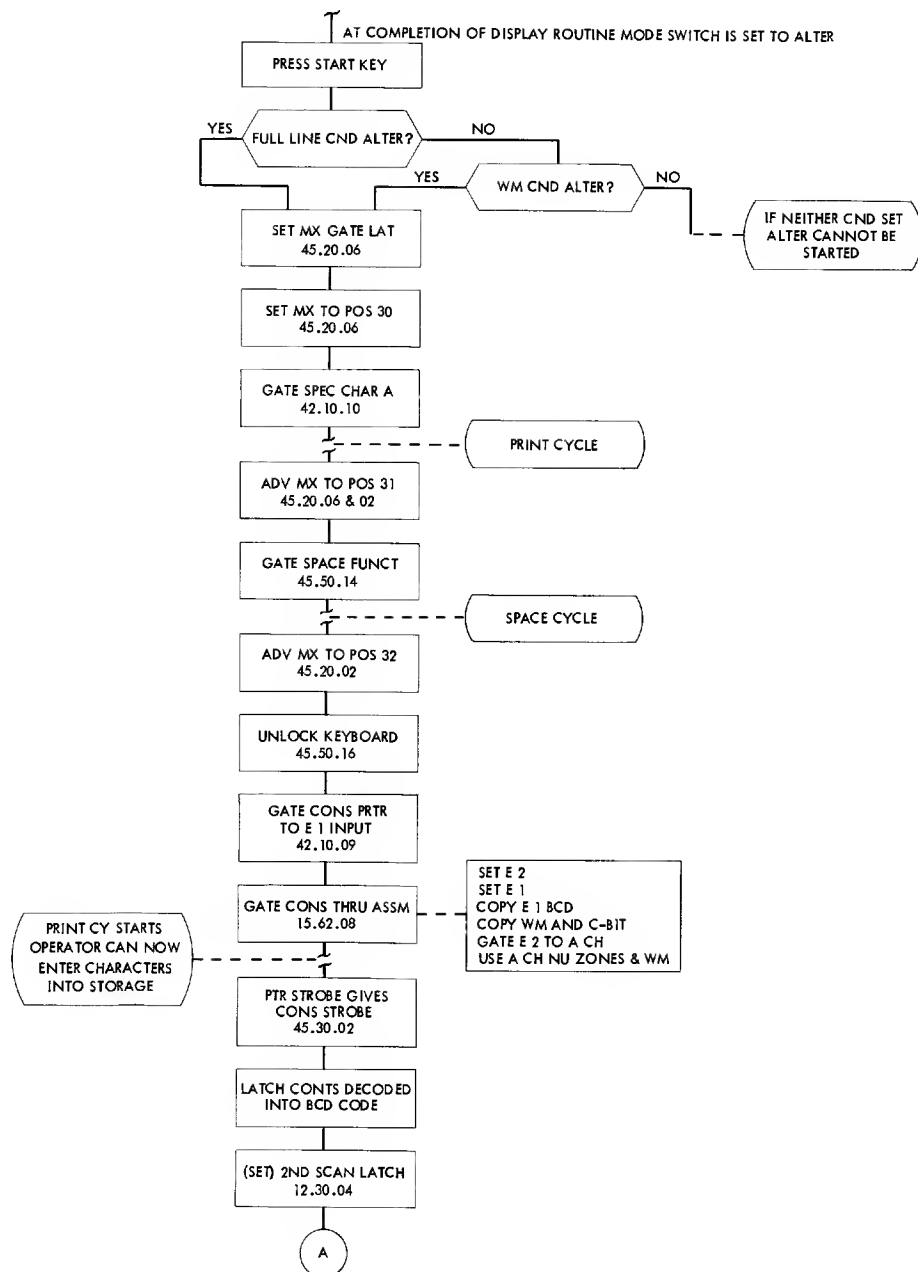
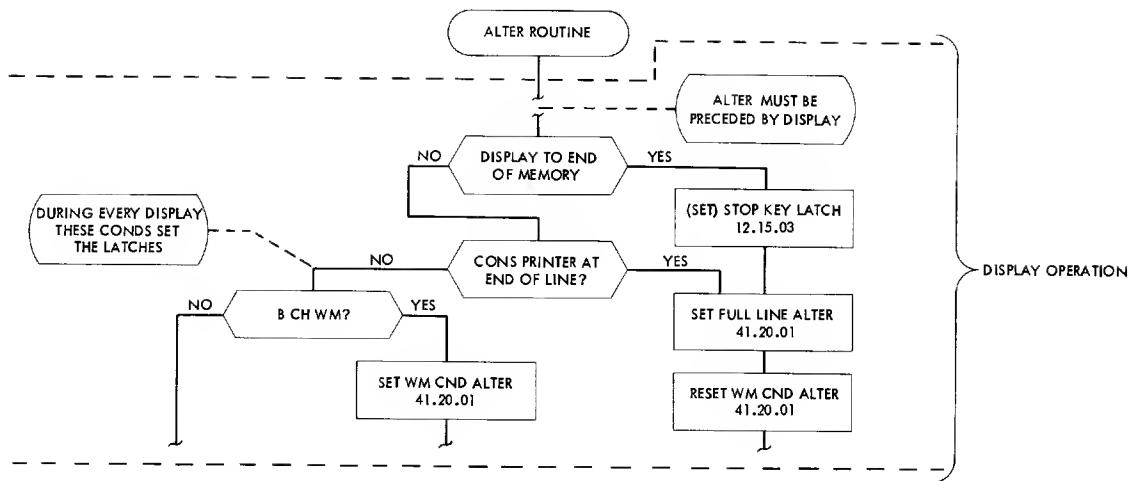


Figure 7.1-11A Alter Routine

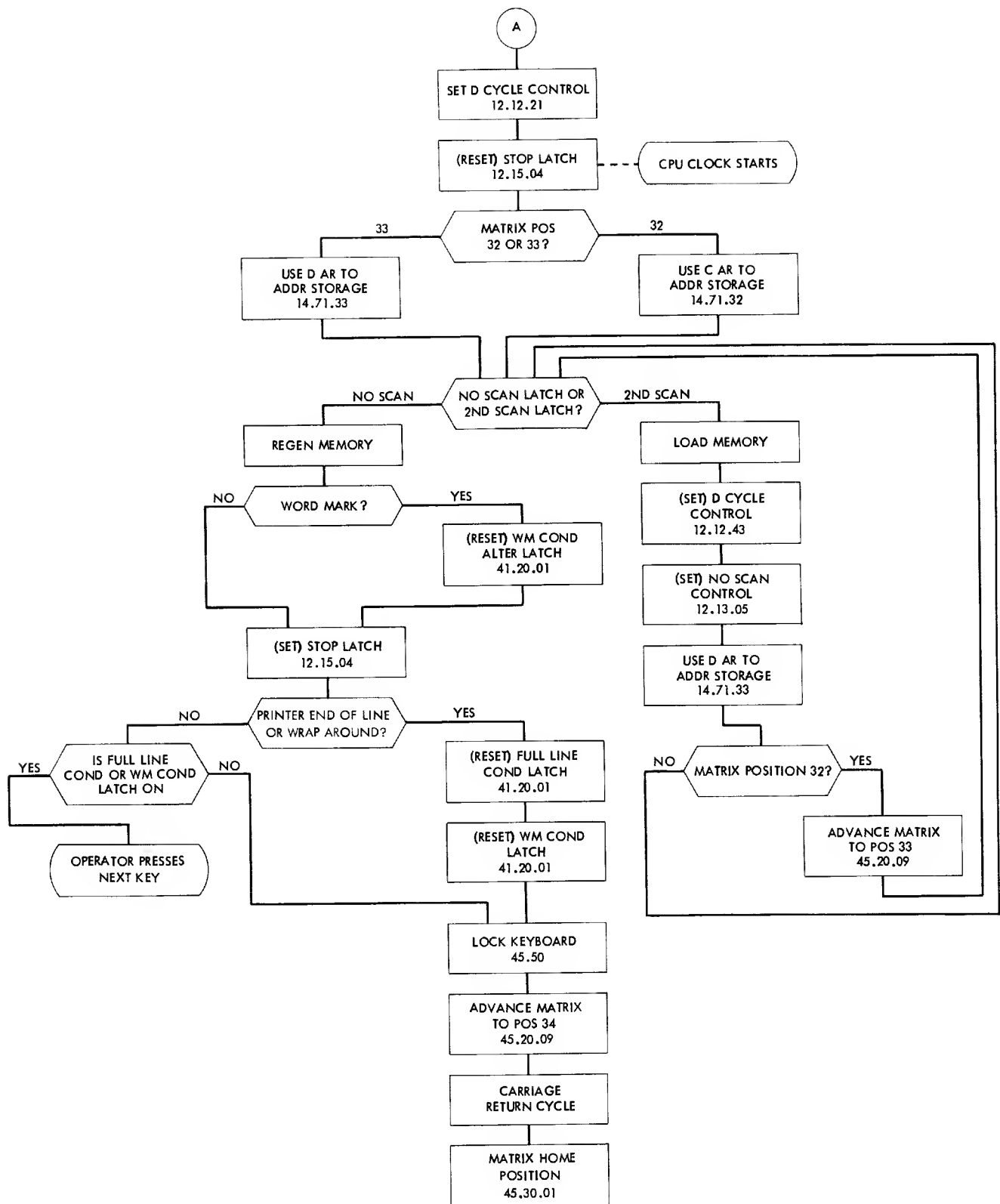


Figure 7.1-11B Alter Routine



a space cycle occurs to advance the matrix to position 32. The keyboard unlocks when the matrix is in position 32 or 33 during an alter operation. With the keyboard unlocked, characters can enter in storage from the input-output printer. When the matrix is in position 32 or 33, the necessary lines are brought up to gate the characters that are keyed in on the printer to the A-channel, through the assembly, to storage. If the first character to be entered in storage has a word mark, the word mark must be keyed in from the printer. Key in word marks to enter them in on the printer. To remove them from storage, do not key them in. While keying in and printing the first character, a console strobe pulse is sent to the CPU. This pulse sets the necessary latches to start the CPU clock and to initiate two consecutive D-cycles. The address used in the first D-cycle is the address that is keyed into the C-address register at the beginning of the display routine. During the first D-cycle, the address reads from the C-address register into the storage address register. It is modified by plus one, and stored back in the D-address register.

While in the storage address register, this address selects the first character in storage to alter and reads it out into the B-data register. This character is replaced with the new character keyed in at the printer. If the character in the B-data register has a word mark, it is not detected at this time, when the control matrix is in position 32. During the second D-cycle the address in DAR reads into the storage address register, and the second character in storage reads out into the B-data register and B-channel.

The address is modified by zero and stored back in DAR. At the end of the second D-cycle, the CPU clock stops and the matrix advances to position 33. The first character in storage is now altered, and the second character is in the B-data register and B-channel.

In matrix position 33 the B-channel is searched for a word mark. If the word-mark-condition alter latch was ON during the display operation, the first B-channel word mark resets this latch and the alter routine line is lowered. One field is then altered. If the full-line-condition alter latch was ON during the display operation, the alter routine line is not lowered until this latch is reset by console printer end-of-line.

In this case one full line is altered. Characters can enter during the alter operation as long as a word mark or end-of-line is not sensed. This depends on which latch was set during the display routine. The console strobe pulse for each character that is keyed in from the printer, starts the CPU clock and causes the CPU to take two D-cycles. The first D-cycle replaces the character in storage with the character that is keyed in. The second D-cycle is used to read-out the next character in storage into the B-data register and B-channel, and to check it for a word mark.

When a word mark or end-of-line is detected, the alter routine line drops and the keyboard locks. After the keyboard locks, the matrix advances to position 34 where a carriage return starts. When the carriage return ends, the matrix returns to home position and the operation ends.

The alter routine can be ended at any time by rotating the mode switch out of the alter mode. Any error, other than a data error, ends the alter routine and no print-out occurs.

### 7.1.15 I/E Mode

The console mode switch in the I/E position gates every last instruction RO cycle and last execute cycle to set the stop latch (12.15.04). Press the start key to start the next phase. A stop print-out is taken after every phase if the print-out control switch is in the normal position.

### 7.1.16 CE Controls

The CE controls on the 1415 console are designed as diagnostic aids for the customer engineer. However, they are available for customer usage for debugging programs and locating errors. To indicate that one of the controls is ON, an off-normal light on the console-indicator-panel lights when one of the CE controls is ON. The controls and settings that cause the off-normal light to come on are:

<u>Control</u>	<u>Setting</u>
Address entry	(not) Normal position
Storage scan	(not) Off position
Cycle control	(not) Off position
Check control	(not) Stop normal position
Print-out control	Inhibit
Asterisk insert	Off

This section covers storage scan, cycle control, check control, check test, start print-out, asterisk insert, and bit switches. Other switches on this panel are covered elsewhere as follows:

<u>Switch</u>	<u>See Section Entitled</u>
Address entry	<u>7.1.12 Console Address Set</u>
Compatibility	<u>8.0.00 IBM 1401 Compatibility</u>
1401 I/O Check reset	<u>IBM 1401 I/O Error Conditions</u>
I/O Check stop	<u>IBM 1401 I/O Error Conditions</u>
Sense switches	<u>Branch If Indicator On</u>
Print-out control inhibit	<u>7.1.09 Stop Print-Out</u>

### Cycle Control Switch (Logic 44.10.01)

The cycle control switch operates only in the Run, I/E Cycle, or CE positions of the mode switch. The setting of the cycle control switch defines how far the CPU clock circuit advances each time the start key is pressed. When set to the logic step position, pressing the start key advances the logic clock by single logic gate pulses. If the storage cycle position is selected, one complete cycle (I, A, B, C, D, E, F, or X) is taken each time the start key is pressed. If the cycle control switch is set OFF, the normal CPU circuits control the logic clock.

At the end of each cycle (storage cycle), or at the end of each logic gate (logic step), the clock stops. When the clock stops, a stop print-out occurs, unless it is inhibited. The character C prints to identify the print-out. If the machine is in the Run or I/E cycle mode and an I/O operation is encountered, the clock runs until the end of the I/O operation.

Pressing the start key resets the stop latch. If the cycle control switch sets to logic step, the stop latch sets on the next 2nd-clock pulse (logic 12.15.04). The clock advances only one logic gate pulse. If the cycle control switch sets to storage cycle, the stop latch sets on the 2nd-clock pulse of logic gate Z, or at the end of the storage cycle (logic 12.15.04). The console stop print-out operation is started by the console stop print-out condition line on logic 12.15.04.

## Check Test Switches

The three check test switches serve as a convenient way to test the checking circuits in the CPU. Three switches, and therefore three tests, are necessary to test all the transistors in the checking circuits.

For each of the three tests, a start program resets. Then press one of the check test switches, and while holding it down, press the start key. The stop latch resets, the clock starts, and the CPU takes an I-cycle. A master error should result, and the CPU should stop with all process system check lights ON. If any check light fails to come on during any of the three check test cycles, there is a failure in the associated checking circuitry.

Errors are forced in the various check circuits in the following manner.

### 1. B-register reset check:

- 1st check test - flips 1st trigger-check trigger (logic 18.14.06)
- 2nd check test - flips reset mem-data-reg trigger (logic 18.14.06)
- 3rd check test - flips reset mem-data-reg trigger (logic 18.14.06)

### 2. Op register set check:

- 1st check test - flips check Op-reg trigger (logic 18.14.04)
- 2nd check test - flips set op-reg trigger (logic 18.14.04)
- 3rd check test - flips set op-reg trigger (logic 18.14.04)

### 3. Op modifier set check:

- 1st check test - flips check op-modifier trigger (logic 18.14.05)
- 2nd check test - flips op-mod-reg trigger (logic 18.14.05)
- 3rd check test - flips op-mod-reg trigger (logic 18.14.05)

### 4. A-character select error:

- 1st check test - blocks all gating to A-channel (logic 18.14.01)
- 2nd check test - forces gate F2 data register to A-channel (logic 18.14.01)
- forces gate E2 data register to A-channel (logic 18.14.01)
- 3rd check test - forces gate op mod register to A-channel (logic 18.14.01)
- forces gate A-data register to A-channel (logic 18.14.01)

### 5. B-character select error

- 1st and 2nd check test - forces channel select error check 1 & 2 (logic 35.10.03)
- blocks load and regen memory (logic 12.50.01)
- 3rd check test - blocks load and regen memory (logic 12.50.01)

#### 6. A-channel validity check

- 1st check test - forces 1 · 8 even and 1 · 8 odd (logic 18. 11. 01)
- 2nd check test - forces numeric-odd and numeric-even (logic 18. 11. 01)
- 3rd check test - forces numeric C-bit and not numeric C-bit (logic 18. 11. 02)

#### 7. B-channel validity check

- 1st check test - blocks VC group one and VC group two (logic 14. 47. 01)
- 2nd check test - forces numeric-odd and numeric-even (logic 18. 12. 03)
- 3rd check test - forces zone · WM · C-bit-odd and zone · WM · C-bit-even (logic 18. 12. 03)

#### 8. Assembly channel validity check

- 1st check test - forces 1 · 8 odd and 1 · 8 even (logic 18. 13. 01)
- 2nd check test - forces numeric-odd and numeric-even (logic 18. 13. 01)
- 3rd check test - forces zones-odd and zones-even (logic 18. 13. 02)

#### 9. Address channel validity check

- 1st check test - blocks VC group one and VC group two (logic 14. 47. 01)
- 2nd check test - forces VC group one and VC group two (logic 14. 47. 01)
- 3rd check test - sets address channel error latch (logic 18. 14. 03)

#### 10. Address exit validity check

- 1st check test - blocks AR-channel group one and two (logic 14. 16. 03)
- 2nd check test - forces AR-channel VC group one and two (logic 14. 16. 03)
- 3rd check test - gates on address exit error trigger (logic 18. 14. 02)

#### 11. A-register set check

- 1st check test - flips reset A-data-reg trigger (logic 18. 14. 07)
- 2nd check test - flips set A-data-reg trigger (logic 18. 14. 02)
- 3rd check test - flips set A-data-reg trigger (logic 18. 14. 02)

### Check Control Switch

The check control switch is a three-position rotary switch. The operations that the check control switch selects are: stop normal, restart, and reset and restart.

1. The stop-normal setting causes an immediate stop on any master error. Error print-out is under control of the print-out control switch.
2. The restart setting stops the operation on any master error, and causes an error print-out the same as stop normal. However, after the print-out, the program automatically restarts (13. 42. 10). If the print-out control switch sets to inhibit, the program restarts immediately after a stop that is caused by an error.
3. When set to Reset And Restart, any error also results in an immediate stop in the same manner as the restart setting. A computer reset operation (13. 42. 10) follows an error print-out operation. When the reset operation completes, the program restarts. If the error print-out is inhibited, Computer Reset and Program Start follow the stop.

## Asterisk Insert Switch

When the asterisk insert switch is ON, and an input device character of incorrect parity is detected on the A-channel, an asterisk enters into the addressed position of storage. The normal A-channel validity-check stop is disabled.

When the switch is OFF, an incorrect parity character from an input device causes an A-channel validity error that stops the CPU (master error).

## Operation

When the A-channel VC circuit indicates the A-channel character is invalid, and the asterisk insert switch is ON, the assembly is controlled to Set Asterisk (15.49.06). This generates an asterisk. The In Cy GM·WM Ctrl blocks the set asterisk control when the position in storage is a GM·WM.

## Storage Scan

The storage scan function is primarily an aid to analyze the 1411 storage area for a read-in or regeneration failure, and to pinpoint the cause of failure.

The method of testing the storage area may be accomplished in four different ways depending on the setting of the storage scan switch. The four settings of the switch are as follows:

1. Regenerate +1. This setting causes the character at the location (that the IAR specifies) to be read out of storage and then regenerates it back into the location from which it was read. The address is modified by plus-one and placed back in IAR. In this manner, each position of storage is scanned, starting at the address keyed into IAR, until pressing the stop key or rotating the mode switch (away from the CE position) stop the operation. Because the highest address of storage is modified, the address modifier produces 00000, and the operation continues from there.
2. Regenerate 0. This mode reads the character out of storage and regenerates it back into storage. Both operations occur at the address that is keyed into the IAR. Because the IAR address is modified by zero, the read-out and regeneration continue repeatedly in the same storage location, until the operation is ended by pressing the stop key or by rotating the mode switch out of the CE position.
3. Load +1. The I-cycles that are taken in this mode read out the character in storage (specified by the address in IAR) and read the bit structure (assembled in the sense bit switches) into the same location in storage. The address keyed into IAR is modified by plus-one and is read back into IAR.

In this mode the same character writes into every storage location, starting with the address keyed into IAR. The IAR address advances until the highest position of storage is read into; then the IAR address starts over at address 00000. The operation continues until stopped by the stop key or by the mode switch (by rotating out of the CE position).

4. Load 0. This mode is the same as Load +1 mode, except that the address in the IAR is not modified by plus-one each time, but remains the same.

#### Operation (Figure 7.1-12)

Start the storage-scan operation by rotating the mode switch to the CE position. With the mode switch in this position, the storage-scan switch is active. The system stops, and a stop print-out occurs when the mode switch is rotated, unless the system was previously stopped. Select the type of storage scan by setting the storage-scan switch to the Load +1 position, Load 0 position, Regenerate 0 position, or the Regenerate +1 position. If a load mode is selected, set the sense bit switches to represent the bit structure to be loaded into storage.

After all the switches are set, pressing the start key, (with the CPU stopped at a last-execute-cycle time and the console control matrix in home position), resets the instruction address register and gates the matrix into position 35. In position 35, the special character # prints to identify this as a storage-scan operation.

After this character prints, the matrix advances to position 36, where a space cycle starts. When the matrix advances to position 1A (X1A-Y1) following the space operation, the keyboard unlocks and the starting address is keyed into the IAR using matrix positions 1A, 2A, 3A, 4A, and 5A the same as in section 7.1.12 Console Address Set.

After the fifth digit, key the address into the IAR. The matrix then advances to position 6A and locks the keyboard. When the keyboard locks, through circuitry it starts the CPU clock. The CPU takes 1-cycles and performs the operation established by the scan switch.

While the matrix is in position 6A, the carriage returns and the matrix advances to home position. If an error occurs during the operation of any one of the four modes of storage scan, the operation stops and a stop print-out occurs. If there is no address modification in the mode, the address in IAR that prints out is the address of the location in storage where the error occurred. If there is an address modification of plus one in the mode, the address that prints out from IAR is one greater than the address of the storage location where the error occurred. After an error occurs and the stop print-out is completed, press the start key to allow the operation to continue.

To end the storage-scan operation at any time, press the stop key or rotate the mode switch out of the CE position. If the print-out control switch is in the normal position, a stop print-out follows.

#### Start Print-Out

The start print-out push-button switch (44.10.01) allows the operator to initiate a stop-print-out routine. The switch is not interlocked so that the operation can be started at any time. The purpose of the switch is to allow the operator to display the contents of the various registers throughout the system, if for any reason the system hangs up in the middle of an operation.

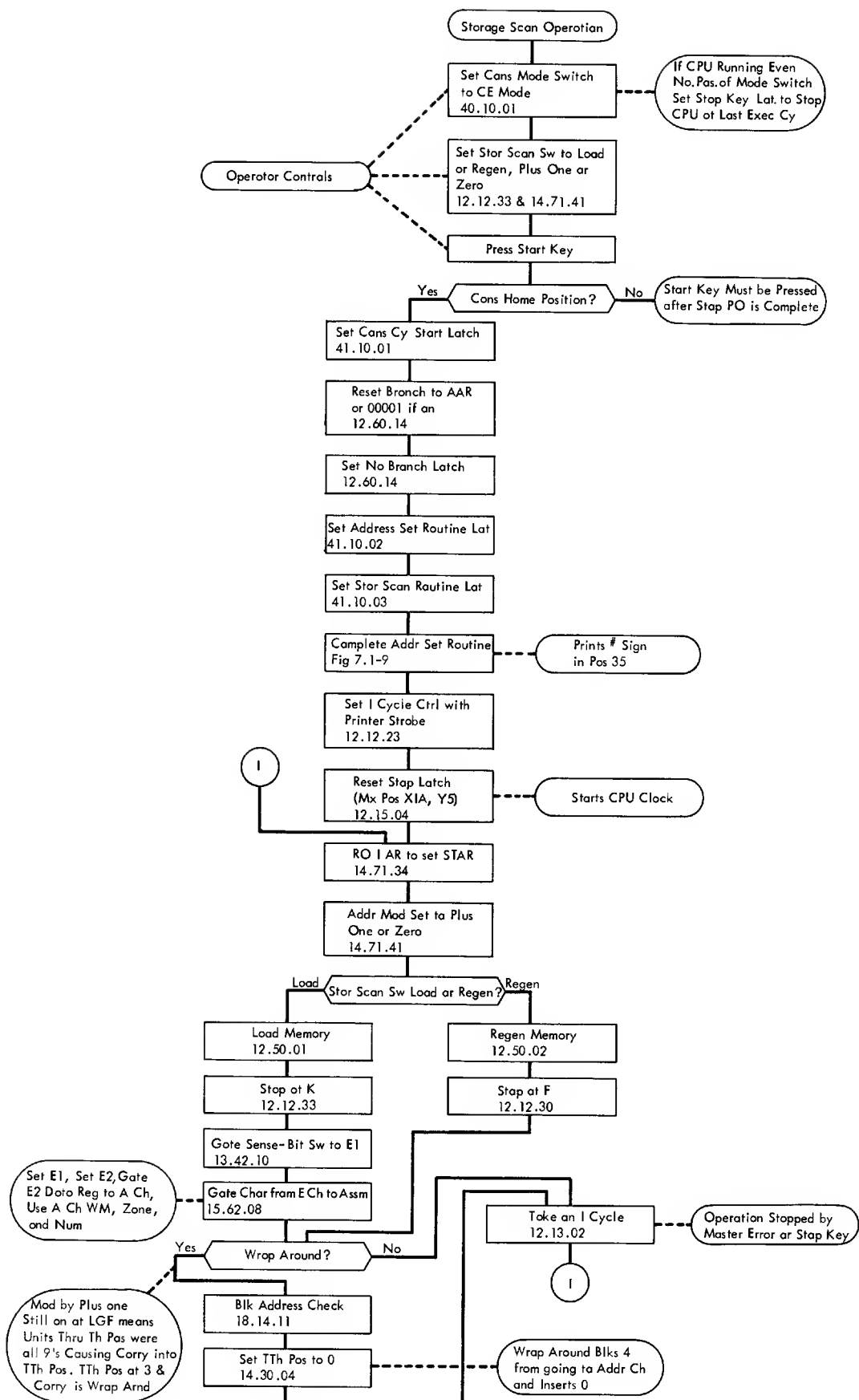


Figure 7.1-12 Storage Scan Operation

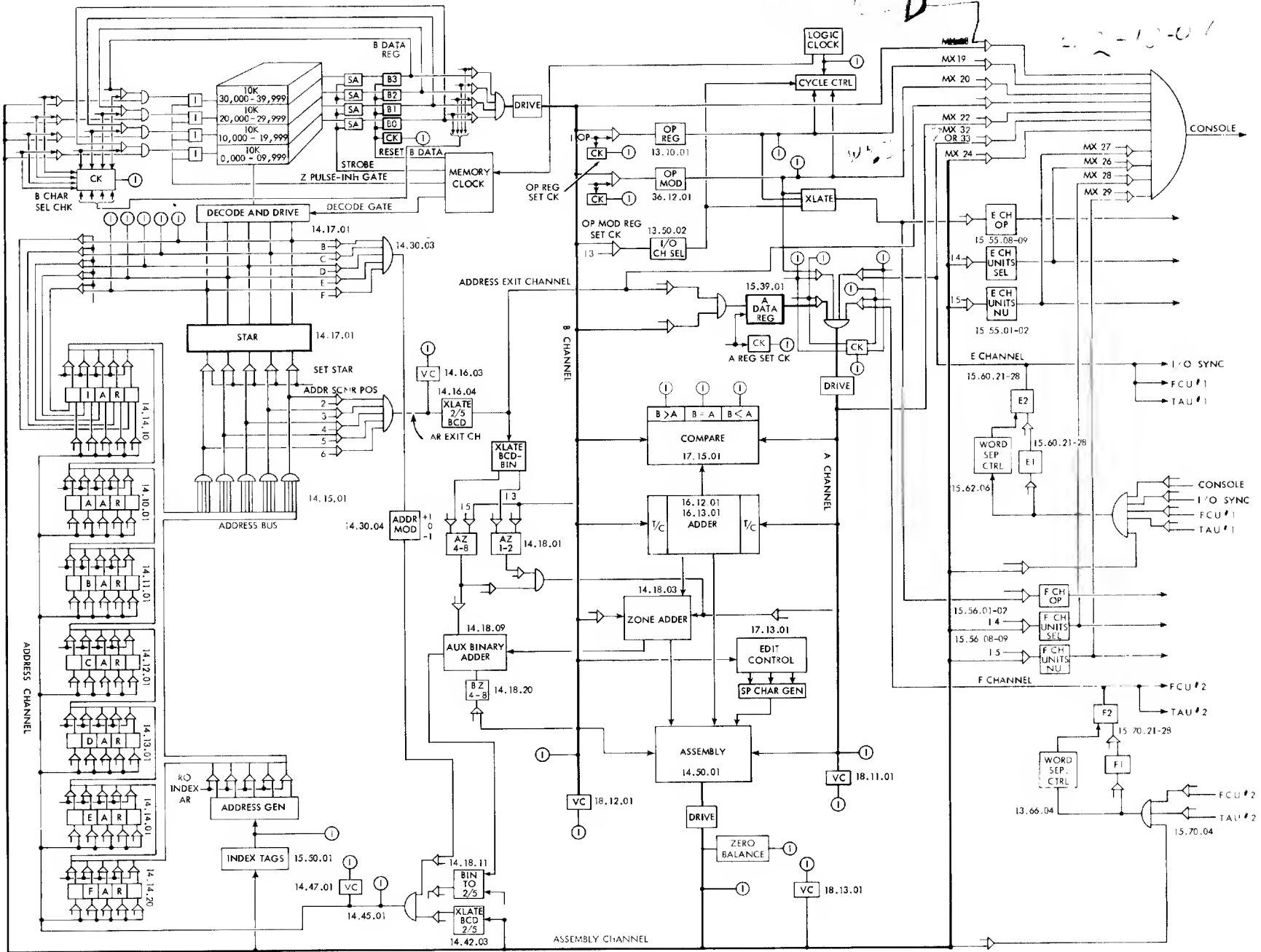


Figure 8.1-1 IBM 1410 Data Flow



## 8.0.00 IBM 1401 COMPATIBILITY

One major difference occurs when the IBM 1410 system operates in the 1401 mode. This is the use of the three-position addresses. The digit portion of the address locates storage positions 000 through 999. The zones over the units and hundreds positions are assigned binary values to increase the number of addresses to 15,999. The A-zone over the hundreds position has a digit value of 1 or 1,000. An address T33 (T is A, 2, 1 bits) locates storage position 01333. The digit value of the other zones are:

B = 2		B = 8
A = 1		A = 4
Hund	Tens	Units

The address to locate position 15,999 is I 9 I (I is A, B, 8, 1 bits).

Some 1401 operations require the addition of two addresses. To handle these operations on a 1410 system requires the addition of the zone adder and the auxiliary binary adder.

### 8.1.00 ZONE AND AUXILIARY BINARY ADDERS

The zone adder combines the binary one- and two-bits while the auxiliary binary adder combines the four- and eight-bits. For example:

	<u>Aux Bin Addr</u>	<u>Zone Adder</u>
	8 4 2 1	
Add A -	0 1 1 1 = 7	
B -	0 1 1 0 = 6	
	0 0 0 1	
Carry	1 1	
	1 1 0 1 = 13	

Notice that each position must be able to carry into the next position. On the data flow diagram (Figure 8.1-1) the line from the zone adder to the auxiliary binary adder is the zone adder carry. In adding two addresses together, there can be a carry from the hundreds position digits that are added in the adder. For example:

		B	
		A	A
A-field	0 1 1 1	8 2 1	= 0 7 8 2 1
		A	A
B-field	0 1 0 1	3 8 0	= 0 5 3 8 0
	0 0 1 0	1 0 1	
	1	1 1 1	
	1		B
		A	A
	1 1 0 1	2 0 1	= 1 3 2 0 1



Again on the data flow diagram (Figure 8.1-1) the line from the adder to the zone adder is the adder carry.

#### Operation (Figure 8.1-2)

When the zones are available to the zone adder, the A-zones combine together and the B-zones combine together. A carry from the A-zones feeds to the B-zones. A carry into the zone adder combines with the A- and B-zones to control the output of the zone adder. The output of the zone adder is the two zones, plus a zone check bit that maintains odd parity. Two B-zones or a carry into a B-zone causes a zone adder carry that feeds to the auxiliary binary adder. The auxiliary binary adder combines the units-positions zones and the zone-adder carry. The output of the zone adder feeds to the assembly unit, and the output of the auxiliary binary adder feeds to the compatibility translator BCD, to 2/5.

#### 8.2.00 INSTRUCTION READ-OUT (FIGURE 8.2-1)

Instruction read-out of a 1401 instruction uses the zone and auxiliary binary to convert the three-position addresses to the five-position addresses that are used by the 1410 to address storage by the 1410.

The instruction word is reverse-scanned (+1) with the Op code as the first character read out of storage. The Op code must contain a WM to be set into the Op register. As in the 1410 mode, a failure to have a WM prevents the next I-cycle and causes an instruction check. The character in the Op register is decoded to control the remainder of instruction read-out (Figure 8.2-2).

The I-ring advances to I1 during the second I-cycle that sets the Spl Adv Ctrl latch to advance the I-ring to I3. Every character read out during instruction read-out sets into the A-data register.

Op codes M, L, D, P and Y are 1401 data move Op codes. When one of these is decoded in the Op decode, it causes the Op modifier register to set with the d-modifier character that is used by the corresponding 1410 data move Op code.

For example, a 1401 D-Op code moves the numerical portion of a single A-field character to the B-field. At I-ring-2-time (LGC of the 2nd I-cycle) this Op code sets the Op modifier register with a 1-bit only. All M or L-Op codes are assumed to be 1401 data move Op codes at I2, and they set the Op modifier register accordingly. At I3, if these Op codes are I/O Op codes, the percent latch sets and the Op modifier register resets with the normal Op modifier character (R/W). For all Op codes except 1401 data move Ops, the Op modifier register sets with every character except the one at the last instruction RO cycle.

The zones of the character that reads out at I3 time are stored in the auxiliary binary registers A1 and A2. The digit is gated to the address channel where it is set into the A and C address registers or the A, B, C, and D address registers (address-double type).

The I-ring advances to I4, and the next character reads out. Any zones over this character set into the hundred position index tag latches. The digits are stored in the address registers.

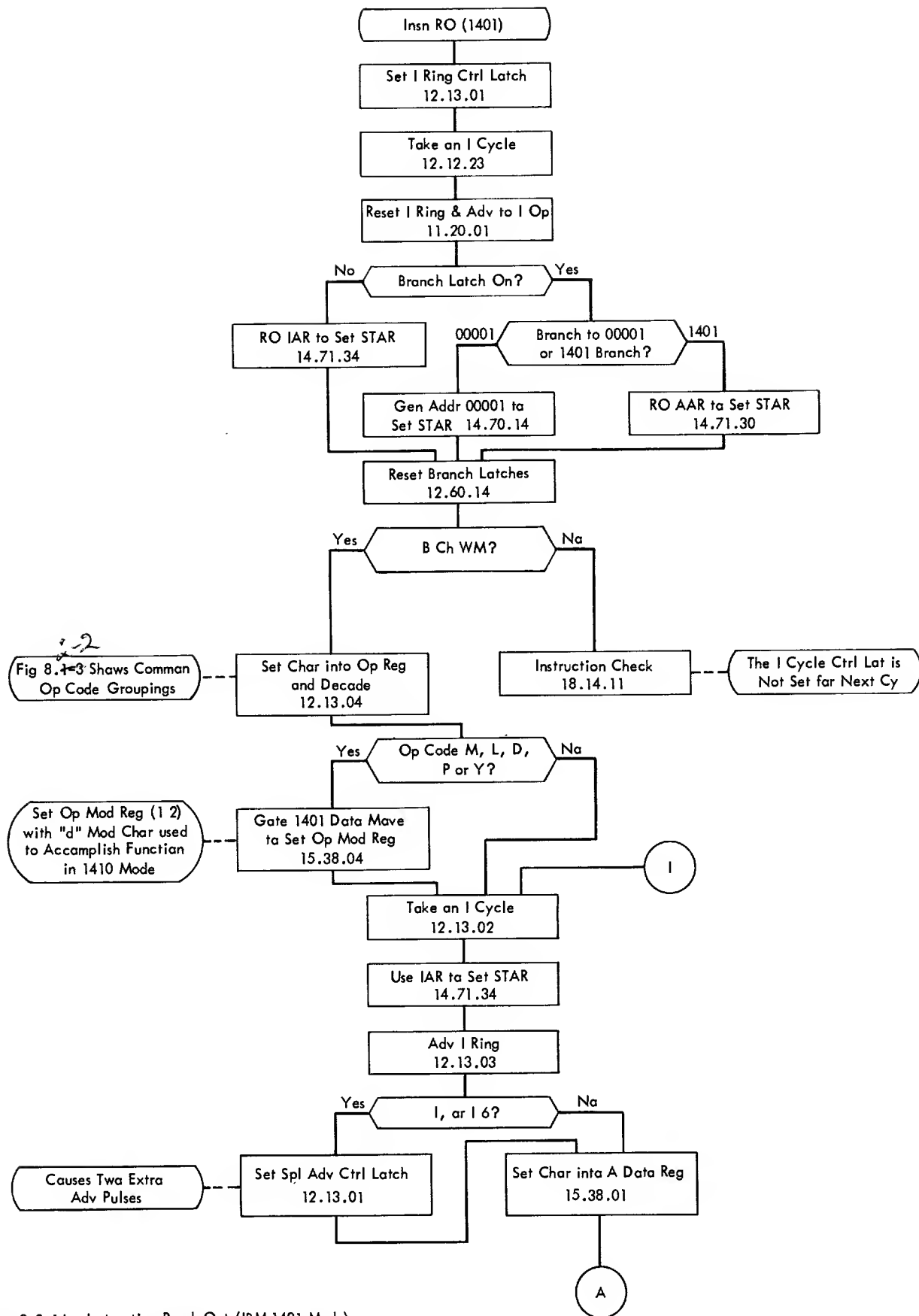


Figure 8.2-1A Instruction Read-Out (IBM 1401 Mode)

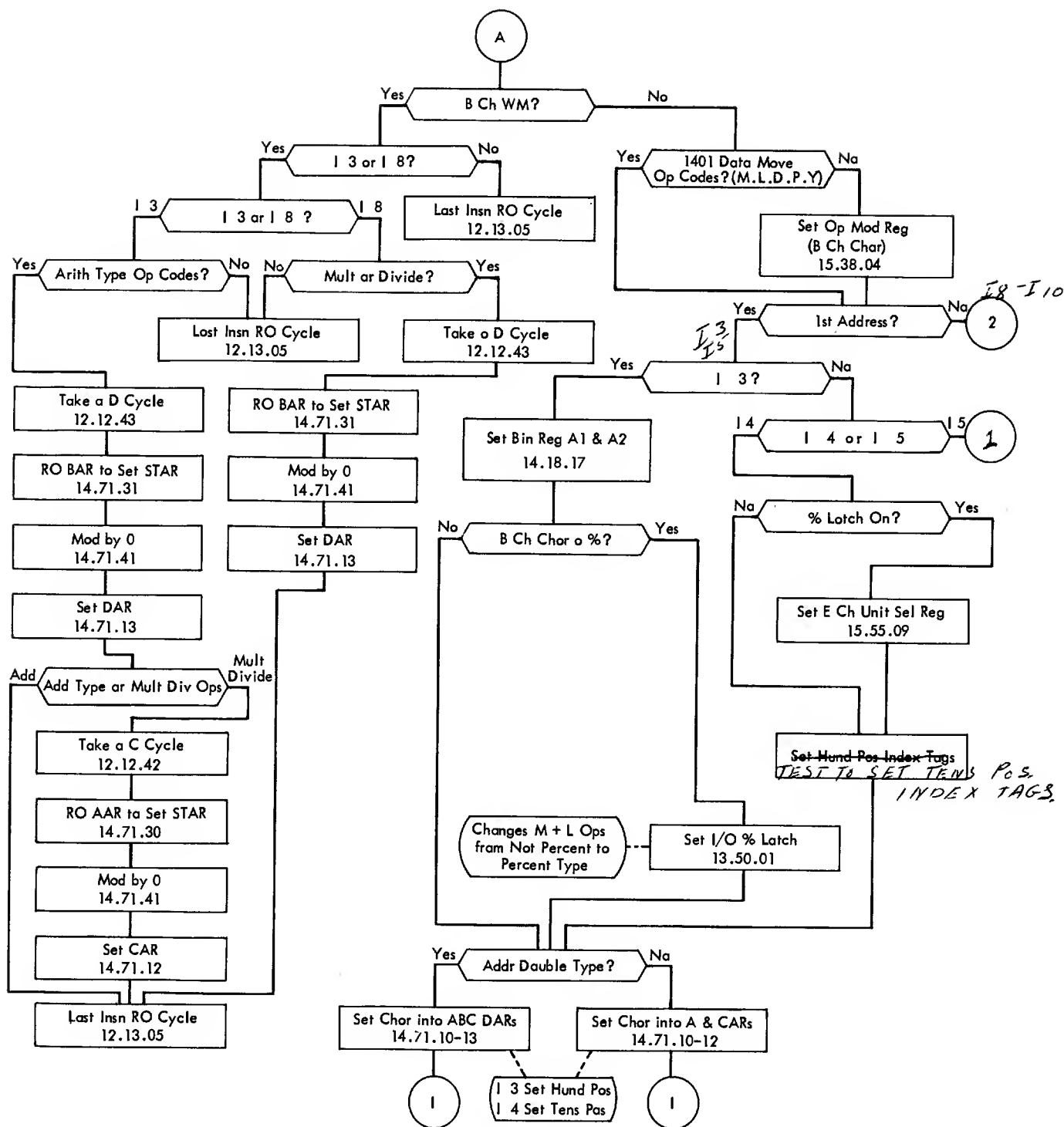
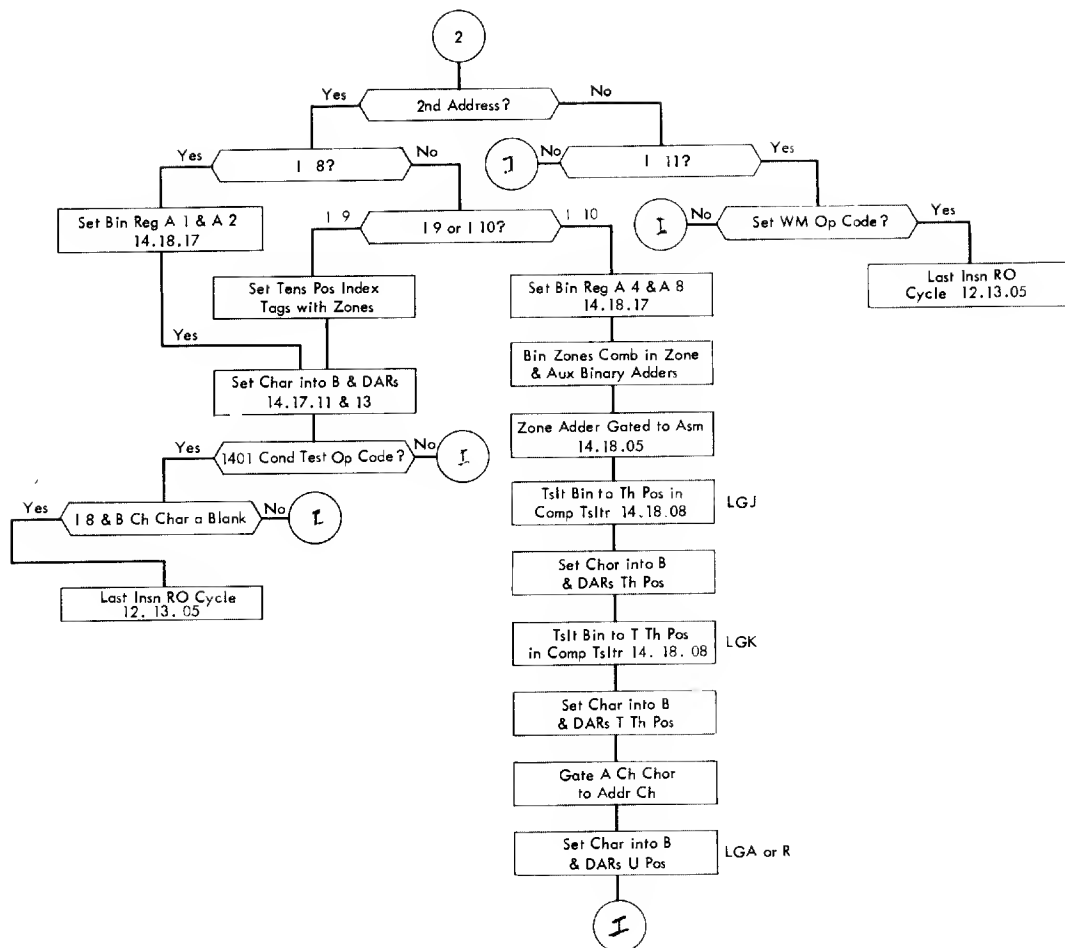
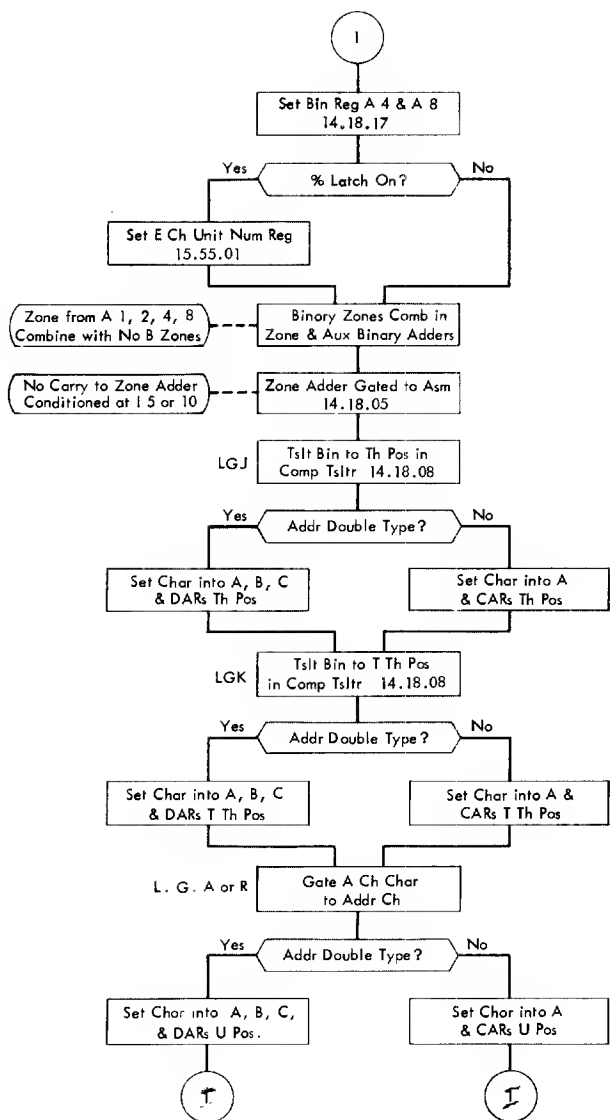


Figure 8.2-1B Instruction Read-Out (1401 Mode)

Figure 8.2-1C Instruction Read-Out (1401 Mode)



# 1401 OP CODES

COMMON OP CODE GROUPING	?	!	A	S	@	%	E	Z	C	W	V	/	.	,	≡	U	#	D	P	Y	B	H	Q	1-7	8	9	M	L	K	F	N
PERCENT TYPE OP CODES																X											%	%			
NOT PERCENT TYPE OP CODES	X	*	X	X	X	X	X	X	X	X	X	X	X	X	X			X	X	X	BS BL						%	%			
ADDR DOUBLE OP CODES	X	X	X	X								X		X	X						BS										
NOT ADDR DOUBLE OP CODES					X	X	X	X	X	X	X		X			X		X	X	X	BL	X	X				%	%	%	X	X
1 ADDR PLUS MOD OP CODES																X					BS										
2 ADDR NO MOD OP CODES	X	X	X	X	X	X	X	X	X			X		X	X																
2 ADDR PLUS MOD OP CODES										X	X							X	X	X	BL						%	%	%		
TWO ADDRESS OP CODES	X	X	X	X	X	X	X	X	X	X	X	X		X	X			X	X	X	BL						%	%	%	%	
ADDR TYPE OP CODES	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X		X	X	X	BS BL						%	%	%	%	
2 CHAR ONLY OP CODES																														X	X
C CYCLE OP CODES					X	X																									
NO C OR D CYCLE OP CODES							X	X	X	X	X	X	X	X	X			X	X	X	BS BL						%	%			
NO D CYCLE AT I RING 6 OPS	X	X	X	X			X	X	X	X	X	X	X	X	X			X	X	X	BL						%	%			
NO INDEX ON 1ST ADDR OPS																X											%	%			
RESET TYPE OP CODES	X	X																													
ADD OR SUBT OP CODES				X	X																										
MPY OR DIV OP CODES					X	X																									
ADD TYPE OP CODES	X	X	X	X																											
ARITH TYPE OP CODES	X	X	X	X	X	X																									
E OR Z OP CODES							X	X																							
COMPARE TYPE OP CODES									X												BL										
BRANCH TYPE OP CODES										X	X	X	X								BS BL										
NO BRANCH OP CODES	X	X	X	X	X	X	X	X	X					X	X	X			X	X	X						%	%	%	X	X
WORD MARK OP CODES														X	X																
M OR L OP CODES																											%	%			
1401 STORAGE AR OP CODES																						X	X								
1401 NO OP LIROC																								X	X						X
1401 CARD OR PRINT																								X							
1ST SCAN FIRST OP CODES	X	X	X	X	X	X	X	X	X	X	X	X		X	X		X				BL	X	X								
A CYCLE FIRST OP CODES	X	X	X	X	X	X	X	X	X					X	X		X	X	X	X							%	%			
STD A CYCLE OP CODES	X	X	X	X	X	X	X	X	X								X	X	X	X							%	%			
B CYCLE FIRST OP CODES										X	X	X									BL										
A REG TO A CH ON B CYCLE OPS	X	X	X	X	X	X	X	X	X			X	X	X	X		X	X	X	X		X	X				%	%			
OP MOD TO A CH ON B CYCLE OPS										X	X					X					BS BL									X	X
LOAD MEM ON B CYCLE OP CODES	X	X	X	X	X	X	X	X									X	X	X	X							%	%			
REGEN MEM ON B CYCLE OP CODES									X	X	X		X								BS BL									X	X
STOP AT F ON B CYCLE OP CODES										X	X	X	X								BS										
STOP AT J ON B CYCLE OP CODES	X	X			X	X	X	X									X	X	X	X							%	%			
READ OUT B AR ON SCAN B CY OPS						X	X	X	X	X	X	X	X	X	X			X	X	X	BS BL						%	%			
READ OUT A AR ON A CYCLE OPS	X	X	X	X			X	X	X					X	X		X	X	X	X		X	X				%	%			

## NOTE

SYMBOL	MEANING	TYPICAL FORMAT
BS	B SHORT	B (I) d
BL	B LONG	B (I) (B) d
M %	M PERCENT	M (% U X) (B) d
M %	M NOT PERCENT	M (A) (B)
L %	L PERCENT	L (% U X) (B) d
L %	L NOT PERCENT	L (A) (B)

Figure 8.2-2 IBM 1410 Common Op-Code Grouping, IBM 1401 Compatibility

The ring is now advanced to I5 where the zones (over the units position of the address) are set into the binary register A4 and A8 latches. The zones from binary registers A1 and A2 are gated to the zone adder where they combine with no-zones on the B-side of the adder. At the same time, the zones from binary registers A4 and A8 combine in the auxiliary binary adder.

The output from the zone adder is gated through the assembly to the 1401 compatibility translator where (along with the output of the auxiliary binary adder) it is translated into the thousands and ten-thousands positions of the address at LGJ and LGK respectively (Figure 8.2-3).

The I-ring advances to I6 where the Spl Adv Ctrl latch again sets to advance the ring to I8. The second address reads out from I8 to I10.

Instruction read-out ends when the B-channel WM over the next Op code is sensed. The set-WM Op code and 1401 condition test Op code end the instruction read-out at I11 and I8 respectively without a word mark to the right of the instruction word.

### 8.3.00 INDEXING

Indexing in the 1401 mode is the same as in the 1410. However, selection of index registers is limited in the 1401 mode because of the use of zone bits over the units and hundreds positions of the three-character 1401 address that represent the thousands and ten-thousands position of the 1410 address. Thus, only the A- and B-bits over the tens position of the 1401 address character can be used to select the index register. The A- and B-bits provide only four combinations. These four combinations select index registers as indicated:

<u>Index Tag</u>	<u>Register</u>	<u>Address</u>
None	—	—
A	13	00087-89
B	14	00092-94
A and B	15	00097-99

The following example represents typical 1401 instruction-word format with index bits:

	A A	A A
	B B	B <del>B</del> <i>B</i>
Op	(1 1 1)	(3 3 3) d

This becomes:

(0 3 1 1 1)	(1 1 3 3 3) d
Indexed by contents of 00097-99	Indexed by contents of 00087-89



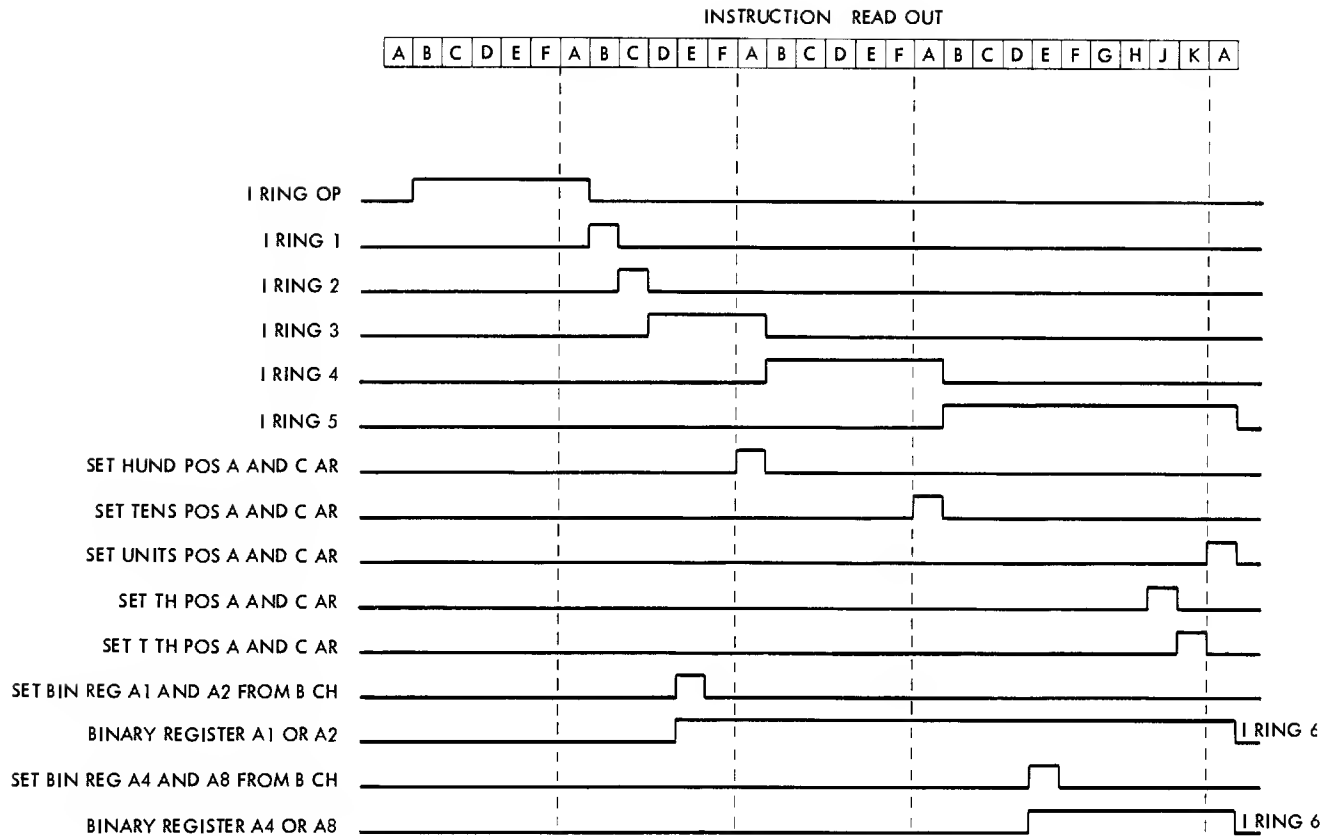


Figure 8.2-3 Zone Adder Timing, Instruction Read-Out

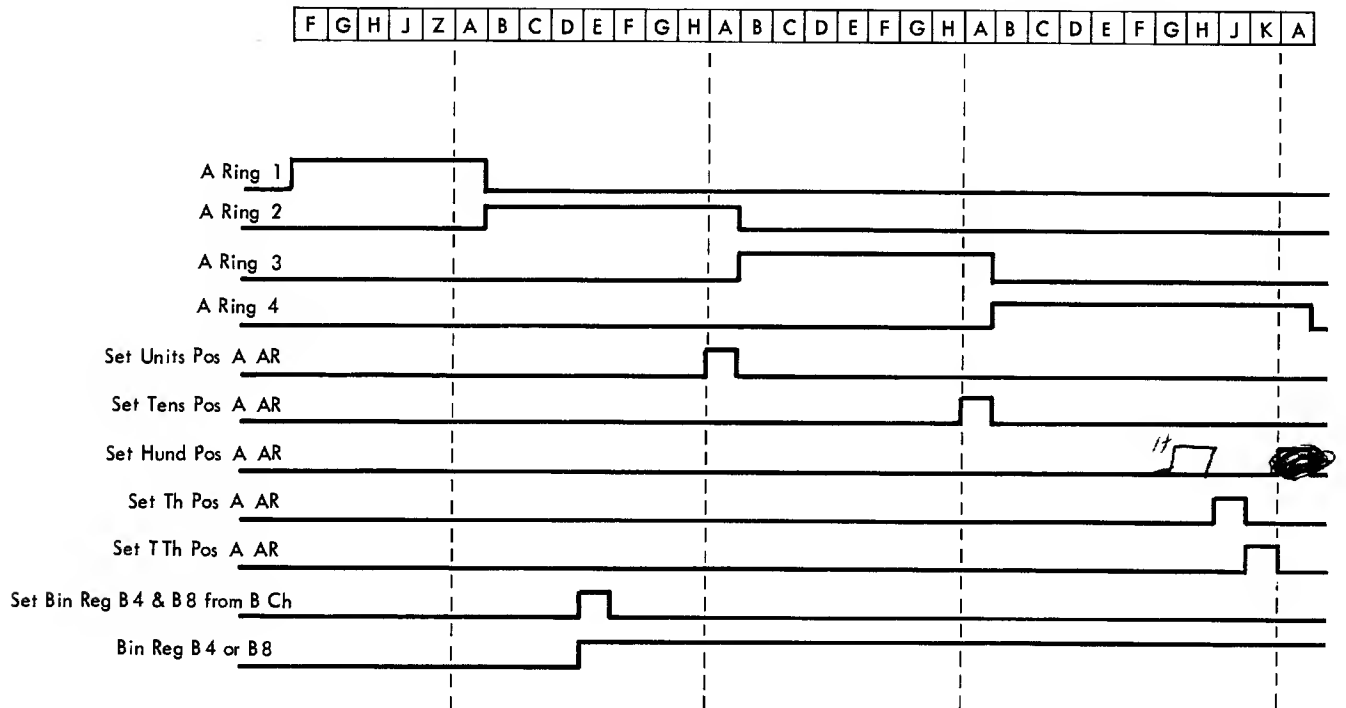


Figure 8.3-1 Zone Adder Timing, Indexing

Instruction read-out begins in the usual manner with the read-out of the Op code from memory. The operation continues as previously explained until I-ring-4 time when the read-out of the tens position detects the index tags. The ten-position index-tag latches set at I-ring-4, LG-A time. This in turn sets the X-cycle control, and starts the A-ring at I-ring-5, LG-F time.

When the I4I0 operates in the 1401 mode, both A- and B-index tags are forced in the hundreds position. This, in combination with the tens-position index tags, selects index registers 13, 14 or 15.

At the end of I-ring-5 time, the A-address has read from memory and converted into a five-numerical digit address in the AAR and CAR. Instruction read-out now temporarily halts to permit indexing of the A-address. The I-ring-5 line remains up until the end of the indexing operation. I6 then turns ON to continue the instruction read-out.

The index tags determine which index register the address generator selects. The address generator sets the STAR. Each position of the index field reads out in sequence to the B-channel starting with the units. This operates the same as 1410 indexing.

The following sequence of events takes place: (Figure 8.3-I)

1. Read out units position of index register. At A-ring-1, LG-A time the address generator sets the STAR to read out the units position of the index register. The zone-bit index tags in the tens position of the A-address determine the address to be generated. The read-out of memory is placed on the B-channel for entry into the adder. Typical contents of an index register in 1401 mode are:

A	A
	B
4	2 2

The zone bits over the units and hundreds position provide the two high-order positions of the index register contents.

2. Set binary register B4 and B8. The zone bits over the units position also read out to the B-channel and set in the B4 and B8 binary register at A-ring-2, LG-E time. These zone bits are added later in the auxiliary binary adder to the zone bits over the units position of the address of the instruction that is being indexed. These zone bits (A4, A8) are still in the A-binary register because the register is not reset until I-ring-6 time of instruction read-out.
3. Reset AAR. The AAR is reset at A-ring-2, LG-F time because the indexed instruction reads into the AAR. The CAR is not reset, but retains the numerical values of the unindexed instruction.
4. Read-out units position instruction character to A-channel. The units character of the CAR (contains the A-address) reads out, translates from 2/5 to BCD, and sets in the A-data register at LG-D time of the first X-cycle.

5. Add units position of the index word to the units position of the instruction word. The adder is now used to add the numerical portion of the units character of the instruction address from the A-data register to the numerical portion of the units character of the index word (from memory but now on the B-channel). The sum is placed in the AAR units position at A-ring-2, LG-H time.
6. Add tens-position characters. At I-ring-3 time the tens-position character from the CAR adds to the tens-position character from the index register. The result sets in the AAR at A-ring-3, LG-H time.
7. A-ring cycling. Only three A-ring times, in addition to A1, are required to index the three-character 1401 program address. The index register and instruction zone bits must add in the zone adder and auxiliary binary adder so that the thousands and ten-thousands characters can set in the AAR. To provide time for this, the A-ring-4 is extended to include LG-J and K.
8. Add hundreds-position numerical characters. At A-ring-4 time, the hundreds-position character reads from the index register to the B-channel. Also during A-ring-4 time, the hundreds-position character from the CAR reads into the A-data register, and then to the A-channel. The result of the hundreds-position addition passes through the assembly area through the BCD to 2/5 translator, and sets in the AAR at A-ring-4, LG-~~A~~ time.  
H
9. Enter zone bits in the zone and auxiliary binary adders. The zone bits from the instruction address are still in the A1, A2, A4, and A8 binary-register latches where they were set during instruction read-out. The zone bits from the index-register units-position character were previously set in the B4 and B8 binary-register latches. The zone bits from the index-register hundreds position (B1 and B2), via the B-channel, are set directly into the zone adder.
10. Add zone bits. The instruction-word zone-bits A1 and A2 add to the index-word zone-bits B1 and B2 in the zone adder. Any carry from the hundreds-position CPU adder-operation also enters in the zone adder. The result of the addition transfers to assembly.  
  
Any carry from the zone adder addition adds in the auxiliary binary adder to the instruction-word zone-bits A4 and A8, and to the index-register zone-bits, B4 and B8.
11. Translate zone-addition output to 2/5 code and set AAR. The result of the zone-adder operation is brought via the assembly area to the compatibility translator. The result of the auxiliary binary-adder operation is also brought to the compatibility translator. The binary results of the zone-bit addition are translated to the 2/5 code. The result represents the value of the indexed thousands and ten-thousands characters of the instruction word. The thousands position sets in the AAR at A-ring-4, LG-J time. The ten-thousands position sets in the AAR at K time. As previously noted, the hundreds position sets at LG-~~A~~ time.  
H
12. Transfer the indexed AAR to the CAR. Indexing is now complete. A C-cycle is taken to read out the indexed A-address of the instruction from the AAR.

This is modified by zero and set in the CAR. I-cycle control is turned ON and the B-portion of the instruction begins with I-ring-6 time.

#### 8. 4. 00 DATA HANDLING

The 1401 D, P, Y, M ( $\overline{Y}$ ) and L ( $\overline{Y}$ ) Op codes perform very nearly the same function as the 1410 D-Op code. There is this significant variation: the 1410 uses different Op modifiers to perform different types of move operations; the 1401 instruction requires a different Op code to obtain variations of the basic move operation.

By comparing the 1410 D-Op code and modifications with the 1401 D, P, Y, M and L operations, it can be seen that in order to duplicate the operation in 1401 mode, it is necessary to generate an appropriate Op modifier. This occurs on the last instruction read-out cycle.

The Op modifier is generated for the various move and load operations in accordance with the following table.

<u>1401 Instruction</u>	<u>d-Character to be Generated</u>		<u>Bits</u>
Move numerical	D (A) (B)	1	$\overline{1}$ , $\overline{2}$ , $\overline{4}$ , $\overline{8}$ , $\overline{A}$ , $\overline{B}$ , $\overline{C}$
Move zones	Y (A) (B)	2	$\overline{1}$ , $\overline{2}$ , $\overline{4}$ , $\overline{8}$ , $\overline{A}$ , $\overline{B}$ , $\overline{C}$
Move character to record, or group-mark with a word-mark	P (A) (B)	.	$\overline{1}$ , $\overline{2}$ , $\overline{4}$ , $\overline{8}$ , $\overline{A}$ , $\overline{B}$ , $\overline{C}$
Move characters to A or B WM	M (A) (B)	C	$\overline{1}$ , $\overline{2}$ , $\overline{4}$ , $\overline{8}$ , $\overline{A}$ , $\overline{B}$ , $\overline{C}$
Load characters to A-field WM	L (A) (B), L (A)	X	$\overline{1}$ , $\overline{2}$ , $\overline{4}$ , $\overline{8}$ , $\overline{A}$ , $\overline{B}$ , $\overline{C}$

The A- and B-bits, along with the 8-bit, determine when the data transfer should end.

<u>Op Modifier Bits</u>	<u>Stop At</u>
$\overline{8}$ , $\overline{A}$ , $\overline{B}$	1st WM—A-field
$\overline{8}$ , $\overline{A}$ , $\overline{B}$	1st WM—A or B-field
$\overline{8}$ , $\overline{A}$ , $\overline{B}$	1st RM or GM, WM, A-field
$\overline{8}$ , $\overline{A}$ , $\overline{B}$	After one position

The 8-bit determines the direction of the scan:

1. 8-bit. Transfer high-to-low order.
2. Not 8-bit. Transfer low-to-high order.

The 1-, 2-, and 4-bits of the Op modifier determine which portion of the A-field characters transfers:

<u>Op Modifier Bits</u>	<u>Use</u>
1	A-field - Num
Not 1	B-field - Num
2	A-field - Zones
Not 2	B-field - Zones
4	A-field WM
Not 4	B-field WM

#### 8.4.01 Move Numerical (D)

##### Op-Code Function

The 1401 D-Op code moves the numerical portion (8-4-2-1) of the single character in the A-address to the B-address. The zone portions (AB) and word marks are not disturbed at either address.

##### Op-Modifier Function

The 1401 D-Op does not have a modifier. The proper 1410 modifier must be set in the Op modifier register to get the desired transfer. The following 1410 functions must be performed:

1. Transfer A-field Numerical (Op modifier bit 1).
2. Transfer B-field WM (Op modifier bit 4).
3. End data transfer after one position (Op modifier bit  $\bar{8} \bar{A} \bar{B}$ ).

##### Operation (Figure 8.4-1)

The Op modifier register sets to  $1 \bar{2} \bar{4} \bar{8} \bar{A} \bar{B} \bar{C}$  to cause the 1410 to function as indicated in the preceding section. Cycling of the 1410 occurs as with a normal 1410 D-Op. The same Op-code grouping lines come up. During the A-cycle, the A-field character reads out of storage and sets into the A-data register. During the following B-cycle, the B-field character reads out to the B-channel. The 1-bit Op-modifier character brings up assembly controls to gate the A-field numerical from the A-channel, and to gate the B-field zone and WM from the B-channel to the core-storage B-field address. The  $\bar{8} \bar{A} \bar{B}$  Op-modifier character ends the operation after one A-cycle and one B-cycle.

#### 8.4.02 Move Zone

##### Op-Code Function

The 1401 Y-Op code moves the zone positions (AB) from the A-address to the B-address. The numerical portion of each character is not disturbed at either address.

##### Op-Modifier Function

The Y-operation code does not exist in the 1410. Therefore, a D-move Op with the proper modifier accomplishes the 1401 Op Y-function in the 1410. The following 1410 functions must be performed. The indicated Op modifier bits must be set.

<u>1410 Functions</u>	<u>Op Modifier Bits</u>
Transfer A-field zones	Set Bit 2
Transfer B-field WM	Set Bit Not 4
End data transfer after one position	Set Bits $\bar{8} \bar{4} \bar{B}$

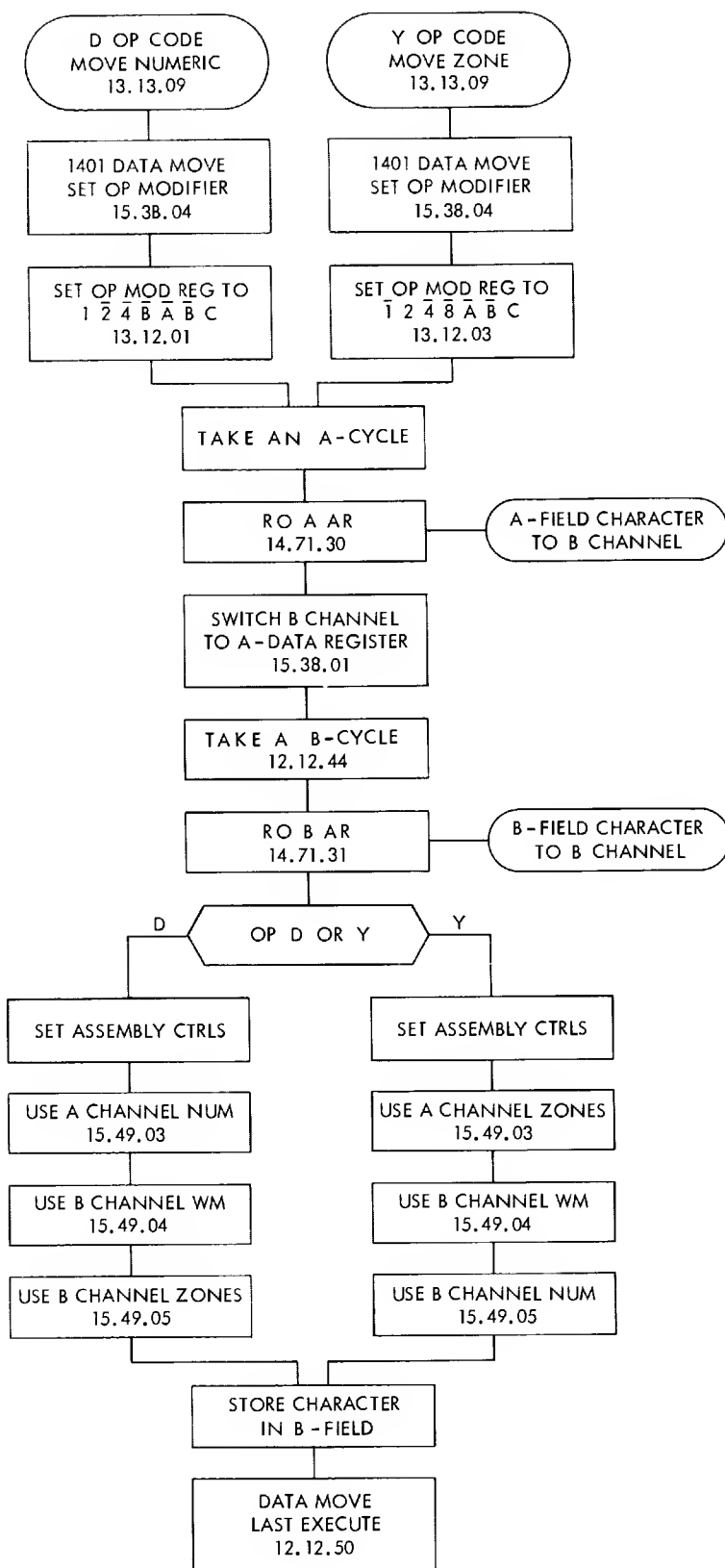


Figure 8.4-1 IBM 1401 Mode, Move Numeric and Move Zone

#### Operation (Figure 8.4-1)

The Op modifier register sets to  $\bar{1} \ 2 \ \bar{4} \ \bar{8} \ \bar{A} \ \bar{B} \ \bar{C}$  to cause the 1410 to function as indicated in the preceding paragraph. The operation continues the same as the move numerical 1401 Op code (see the section entitled 9.2.01 Switches, Keys, and Lights). The only difference is the setting of the Op modifier register and the assembly unit controls.

#### Circuits

Set Op modifier register to  $\bar{1} \ 2 \ \bar{4} \ \bar{8} \ \bar{A} \ \bar{B} \ \bar{C}$ . This operates the same as the move numerical Op code (see the section entitled 8.4.01 Move Numerical (D)). The only difference is that the 2 Op-modifier register bit is set instead of the 1-bit.

#### 8.4.03 Move Characters to Record or Group Mark *WORD MARK*

##### Op-Code Function

The 1401 P-Op code moves an entire record from one core-storage area to another. The A- and B-addresses specify the high-order position of each area of storage. Transmission starts at the high-order address of the A-field and continues until a record mark (A82 bits) or a group-mark with a word-mark (BA 8421 WM) is sensed in the A-field. The record mark or group-mark with a word-mark transfers to the B-field. Word marks in the B-field remain unchanged. A-field word marks do not transmit except for the group-mark with a word-mark combination.

The P-Op code does not exist with the 1410. Therefore, a D-move Op with the proper modifier accomplishes the 1401 Op P-function with the 1410.

##### Op-Modifier Function

The 1401 P-Op does not have an Op modifier. Therefore, the proper 1410 D-Op code modifier must be set in the Op modifier register to get the desired transfer of data. The following 1410 functions must be performed:

<u>1410 Functions</u>	<u>Op-Modifier Bits</u>
Transfer A-field Numerical	Bit 1
Transfer A-field zones	Bit 2
Transfer B-field WM	Bit not 4
Transfer high to low-order	Bit 8
Stop operation with 1st RM or GM · WM A-field	8 A B

#### Operation (Figure 8.4-2)

The Op modifier register sets to  $1 \ 2 \ \bar{4} \ 8 \ A \ B \ \bar{C}$  to cause the 1410 to function as indicated in the preceding paragraph. After this is done, alternate A- and B-cycles perform the operation. See the section entitled 2.8.01 Move Data Op Code (D).

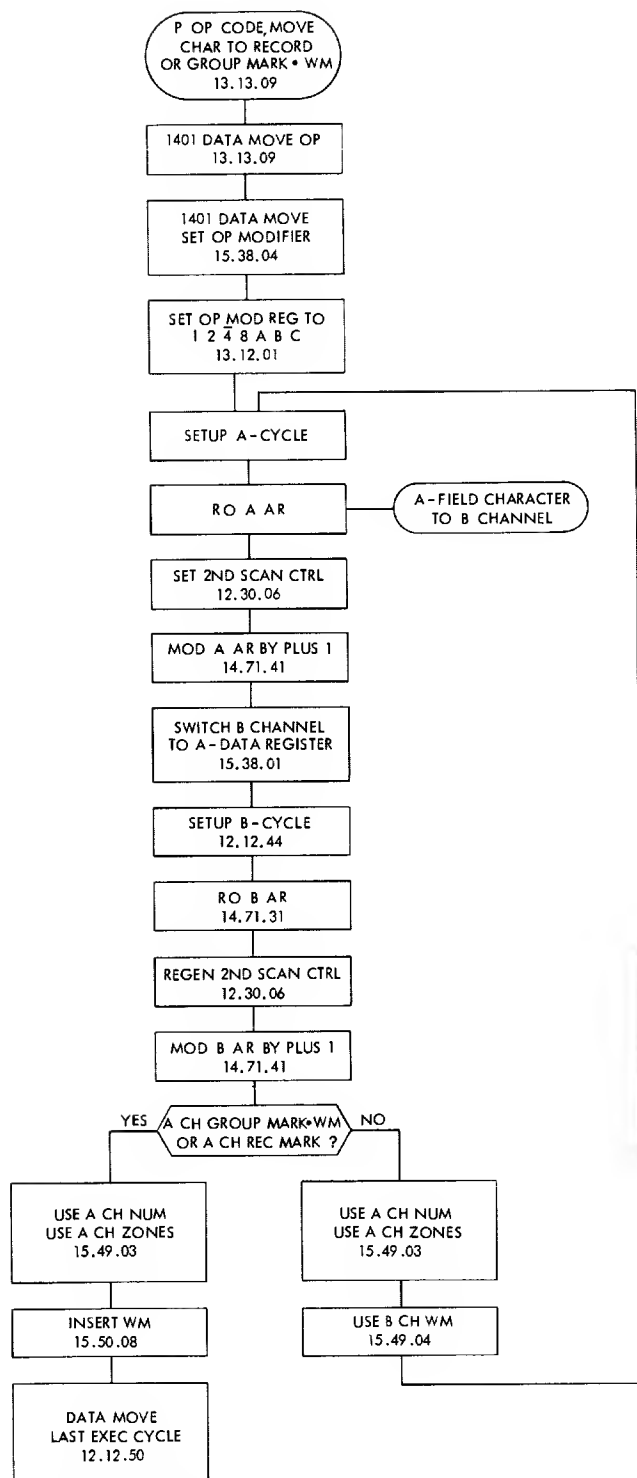


Figure 8.4-2 IBM 1401 Mode, Move Character to RM or GM WM



The 1- and 2-bit lines from the Op modifier reg bring up the assembly controls that gate to storage in the B-field location the desired portions of the A- and B-channel characters with the proper parity. The next character of the A-field reads out on the following A-cycle, and the operation continues by alternate A- and B-cycles, until a WM reads out and combines with the  $\bar{8}$  A B bits from the Op modifier register to end the operation.

#### 8.4.04 Move Characters to A or B Word Mark M (A) (B), M (A)

##### Op-Code Function

The M-Op code moves data in storage from the A-field address to the B-field address. The first word mark encountered in either field stops the operation. In the case of a single address instruction, the B-address is taken from the BAR and does not have to be written or interpreted as part of the instruction. The first WM encountered stops the operation as in the 2 Addr Insn.

##### Op-Modifier Function

The 1401 M-Op does not have a modifier. The appropriate 1410 modifier must be set in the Op modifier register to get the desired transfer. The following 1410 functions must be performed:

<u>1410 Functions</u>	<u>Op-Modifier Bits</u>
Transfer A-field numerical	Bit 1
Transfer A-field zones	Bit 2
Transfer B-field WM	Bit 4
Set first scan	Bit $\bar{8}$
End operation at first WM A or B-field	Bit $\bar{8}$ A B

##### Operation

During the last instruction RO cycle the Op modifier register sets to 1, 2,  $\bar{4}$ ,  $\bar{8}$ , A, B, C to cause the proper 1410 functions as indicated in the Op-Modifier Function paragraph. Also, the controls set up for an A-cycle. The first character of the A-field reads out during the A-cycle, and sets into the A-data register. The next cycle is a B-cycle during which the first character of the B-field reads out and is placed on the B-channel.

#### 8.4.05 Load Characters to A-Word Mark L (A) (B), L (A)

##### Op-Code Function

The L-Op code moves the data and the word mark in storage from the A-field address to the B-field address. The A-field word mark stops the operation. B-field word marks clear unless the B-field is larger than the A-field.

##### Op-Modifier Function

The 1401 L-Op does not have a modifier. The appropriate 1410 modifier must be set in the Op modifier register to effect the desired transfer. The following 1410 functions must be performed:

1410 FunctionsOp-Modifier Bits

Transfer A-field numerical  
 Transfer A-field zones  
 Transfer A-field word mark

Bit 1  
 Bit 2  
 Bit 4

Set the first scan-and-end-operation at first A-field WM: Op mod  $\bar{8}$  A  $\bar{B}$ -bit.

### Operation

During the last instruction RO cycle the Op modifier register sets to 1 2 4  $\bar{8}$  A  $\bar{B}$  C to cause the proper 1410 functions as indicated in the preceding Op Modifier Function paragraph. Also, the controls set for an A-cycle, and the first character of the A-field reads out and sets in the A-data register. The next cycle is a B-cycle during which the first character of the B-field reads out, and is placed on the B channel. The 1-, 2-, and 4-bit lines from the Op modifier register bring up the assembly controls that gate to storage in the  $\bar{B}$ -field location, the numerical, zone and WM of the A-channel characters that have the proper parity. B-field WM, if any, clears from storage.

The next character of the A-field reads out on the following A-cycle, and the operation continues by alternate A- and B-cycles until an A-channel WM combines with the  $\bar{8}$  A  $\bar{B}$ -bits from the Op modifier register to end the operation.

#### 8.4.06 Move Characters and Suppress Zeros Z (A) (B)

The function, operation and circuits of the execute phase of <sup>Z</sup> (A) (B) in 1401 mode are the same as the standard 1410 operation. Note: on 13.13.03, Z-Op decode is common to both operations. See section 2.8.02 Move Character and Suppress Zero Op Code (Z).

#### 8.4.07 Compare C (A) (B)

The function, operation, and circuits of the execute phase of C (A) (B) in 1401 mode are the same as the standard 1410 operation except for set of High-Low-Equal that does not occur in the 1401 mode for single-character C-Ops. C-Op decode (13.13.03) is common to both operations. Section 2.8.03 contains the description of this operation.

#### 8.4.08 Move Characters and Edit E (A) (B)

The Edit Operation in 1401 mode is accomplished exactly as in 1410 program execution. E Op decode (13.13.02) is common. See Section 2.8.04 Edit (E).

### 8.5.00 ARITHMETIC

#### 8.5.01 Add and Subtract

These Op codes operate the same as in 1410 mode except for the sign, zones and overflow controls.

## Sign

If the signs of the fields and type of Op code require a true-add, the zones from the units position of the B-field are used as the sign of the result.

If a complement-add is required, the sign of the B-field is used as the sign of the result but it is inserted in the standard form (8-bit only, minus; AB bits, plus). If a recomplement cycle is required, the sign of the B-field is inverted to a standard sign.

## Zones

All zones in the B-field are removed, except the units position sign or high-order position on true-add.

## Overflow

An adder carry during the B-cycle, in which a B-channel word mark is sensed, indicates an overflow condition. This sets the arithmetic overflow latch, which is testable, by a branch instruction.

The adder carry is also gated to the zone adder, where it is combined with any A- or B-channel zones. The zone adder output is gated through the assembly and is stored in the high-order position of the B-field. For example:

A-field	8 0 2
B-field	2 9 1
	—
	0 9 3
Arith Overflow	←

### 8.5.02 Zero and Add, Zero and Subtract

The adder is not used for these Op codes. The numerical portion of the A-channel is gated through assembly. The sign of the A-field is used on zero and add, and is inverted on zero and subtract. No other zones are used. Zeros are inserted on the assembly channel for all extra, high-order positions of the B-field.

### 8.5.03 Multiply and Divide

These Op codes operate the same as in 1410 mode. The arithmetic overflow latch sets instead of the divide overflow latch.

## 8.6.00 LOGIC OPERATIONS

### 8.6.01 Branch (B) B ( I ) b

#### Op-Code Function

The B-Op code with an I-address and a blank d-modifier is an unconditional branch. This means that the normal program sequence is interrupted and continued at any

desired point, as specified by the I-address, without testing for specific conditions.

#### Operation (Figure 8.6-1)

When a branch is required, the CPU takes a B-cycle. During the B-cycle, the IAR reads out to the STAR through the modifier to the BAR. A modify-by-zero condition exists in the modifier because the no-scan latch is set.

The 1401 branch latch sets to cause the AAR to read out to STAR for the I-Op cycle of the instruction read-out.

This is the start of normal instruction read-out for the first branch routine instruction.

#### 8.6.02 Branch, if Indicator On B ( I ) d

##### Op-Code Function

The d-character modifier specifies the condition to be tested. If the indicator tested is OFF, the next instruction in sequence is taken. If the indicator tested is ON, the program branches to the I-address.

Figure 8.6-2 illustrates the d-modifiers and the indicators they test.

#### Operation (Figure 8.6-1)

At I-ring-8 time, during instruction read-out, the Op-modifier register sets with the Op-modifier character and is decoded.

The Op-modifier decode compares with the indicators and at I-ring-9 time (last instruction RO cycle either the branch or no-branch condition is brought up. No-Branch causes the last execute cycle that initiates the instruction read-out of the NSI. The 1401 branch latch causes the extra B-cycle, as covered in the section entitled 8.6.01 Branch (B)).

#### 8.6.03 Branch, if Character is Equal B ( I ) (B) d

##### Op-Code Function

This instruction code causes the single character at the B-address to compare to the d-character. If the bit configuration is the same, the program branches to the I-address. If not the same, the program continues in sequence. The d-character can be any combination of the six BCD code bits (AB 8421).

#### Operation (Figure 8.6-3)

At I-ring-11 time, during instruction read-out, the Op-modifier register sets with the Op-modifier character. At I-ring-12 time, the 1st scan-control latch sets and a B-cycle starts. The character at the B-field address reads to the B-channel. The Op-modifier register is gated to the A-channel and a comparison is made between A- and B-channels in the compare unit. If the equal latch sets, the program branches to the I-address in the AAR. In all other cases the program continues to the NSI.

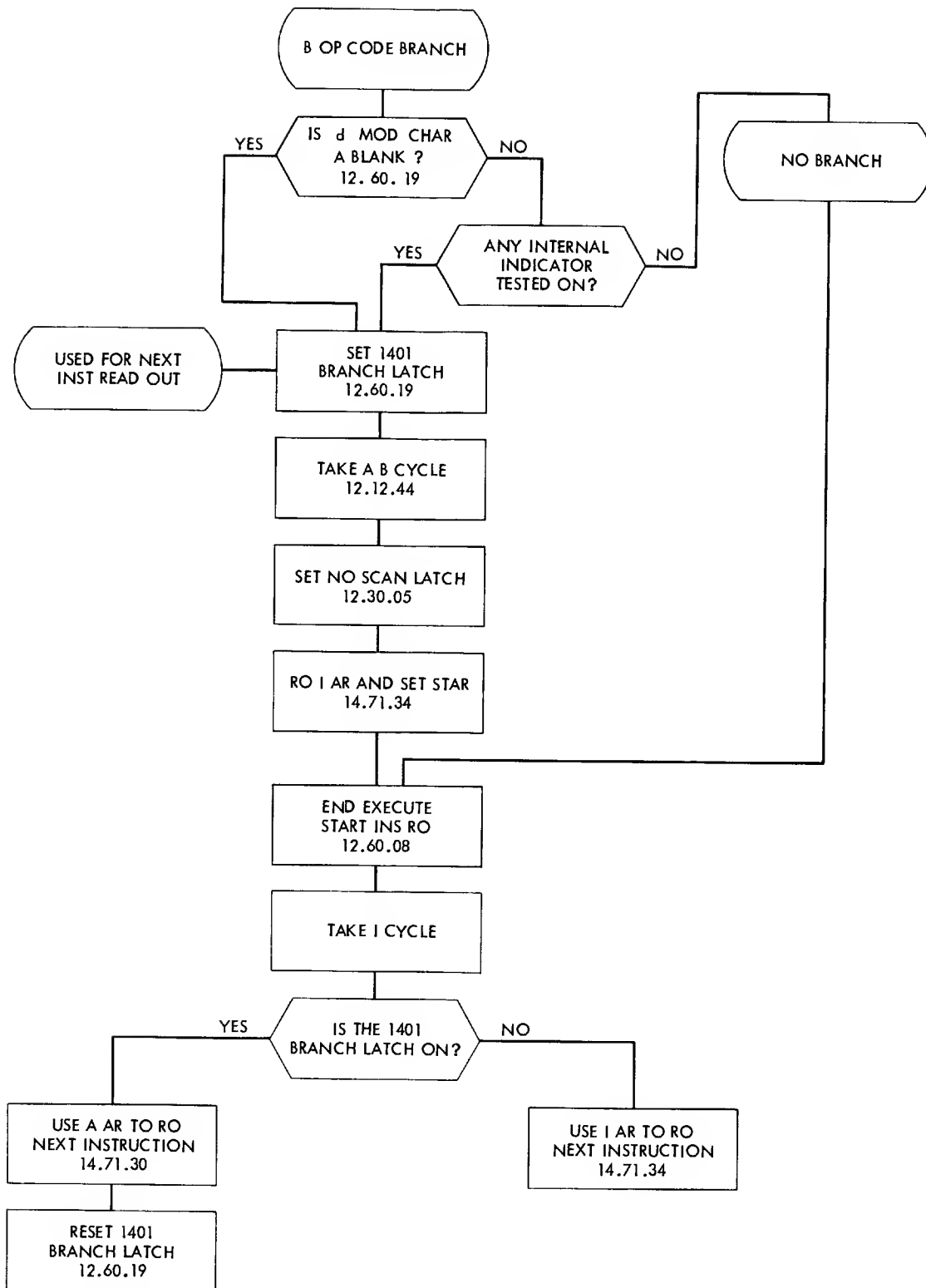


Figure 8.6-1 Branch Unconditional, and Branch If Indicator On

CHARACTER AT d FOR B (I) d BRANCH IF INDICATOR ON					
d	BRANCH ON	LOGIC	d	BRANCH ON	LOGIC
bl	UNCONDITIONAL	12.60.01	S	EQUAL COMPARE B = A	12.60.01
9	CARRIAGE CHANNEL #9	12.60.01	T	LOW COMPARE B<A	12.60.01
A	"LAST CARD" SWITCH	12.60.18	U	HIGH COMPARE B>A	12.60.01
B	SENSE SWITCH B	12.60.18	V	VALIDITY CHECK	12.60.17
C	SENSE SWITCH C	12.60.18	W	FILE WRONG LENGTH REC	12.60.17
D	SENSE SWITCH D	12.60.18	X	FILE ADDRESS COMPARE	12.60.17
E	SENSE SWITCH E	12.60.18	Y	ANY FILE CHECK	12.60.17
F	SENSE SWITCH F	12.60.18	Z	OVERFLOW	12.60.02
G	SENSE SWITCH G	12.60.18	*	INQUIRY CLEAR	12.60.17
K	END OF REEL	12.60.05	?	READER ERROR IF I/O CHK STOP SWITCH OFF	12.60.18
L	TAPE ERROR	12.60.17	!	PUNCH ERROR IF I/O CHK STOP SWITCH OFF	12.60.17
N	ACCESS INOPERABLE		‡	PRINTER ERROR IF I/O CHK STOP SWITCH OFF	12.60.17
P	PRINTER BUSY	12.60.17	@	CARRIAGE CHANNEL #12	12.60.01
Q	INQUIRY REQUEST	12.60.02	%	PROCESSING CHECK WITH PROCESS CHK SWITCH OFF	12.60.18
R	CARRIAGE BUSY	12.60.15	/	UNEQUAL COMPARE B ≠ A	12.60.01

Figure 8.6-2 Character At d For Branch If Indicator On

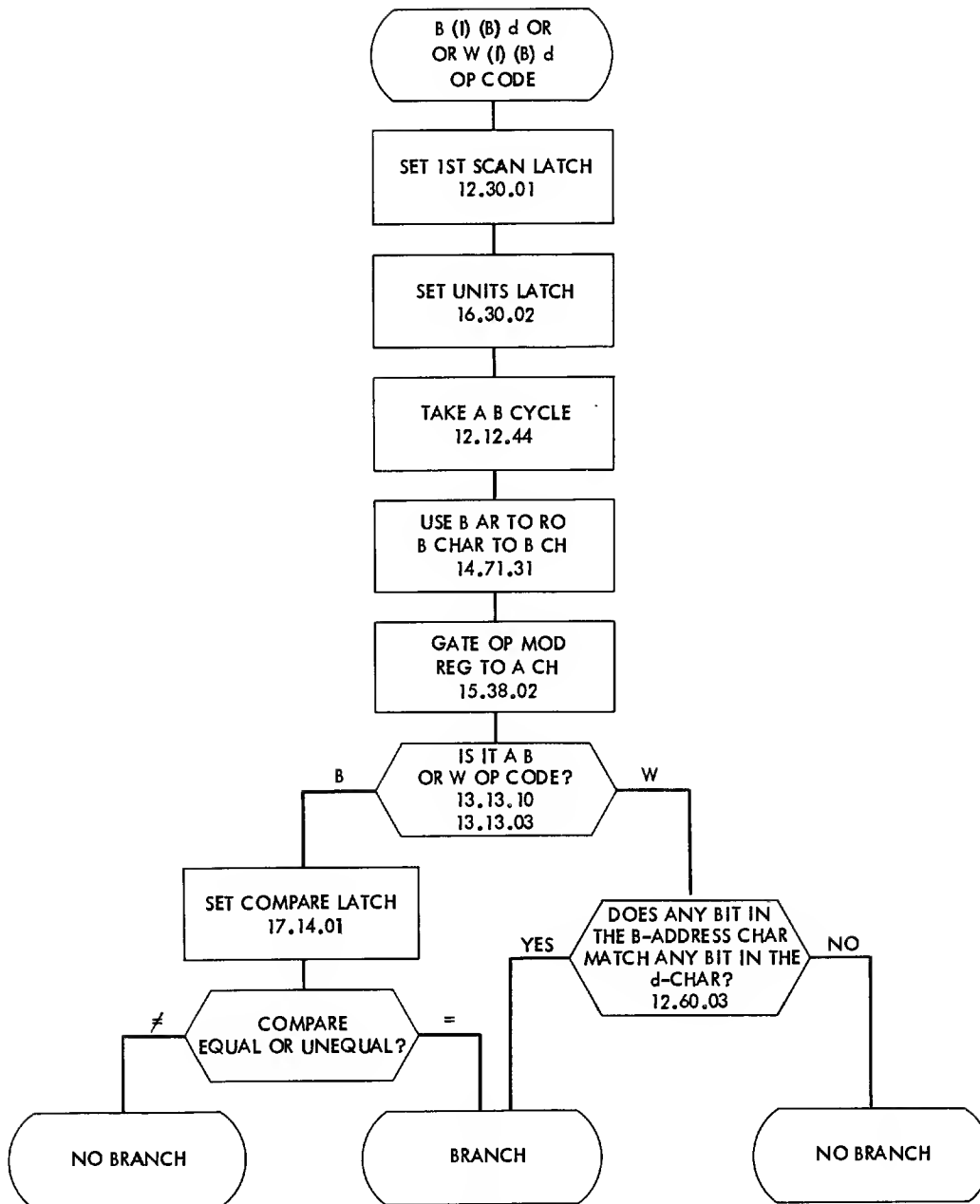


Figure 8.6-3 Branch If Character Is Equal, Branch If Bit Equal

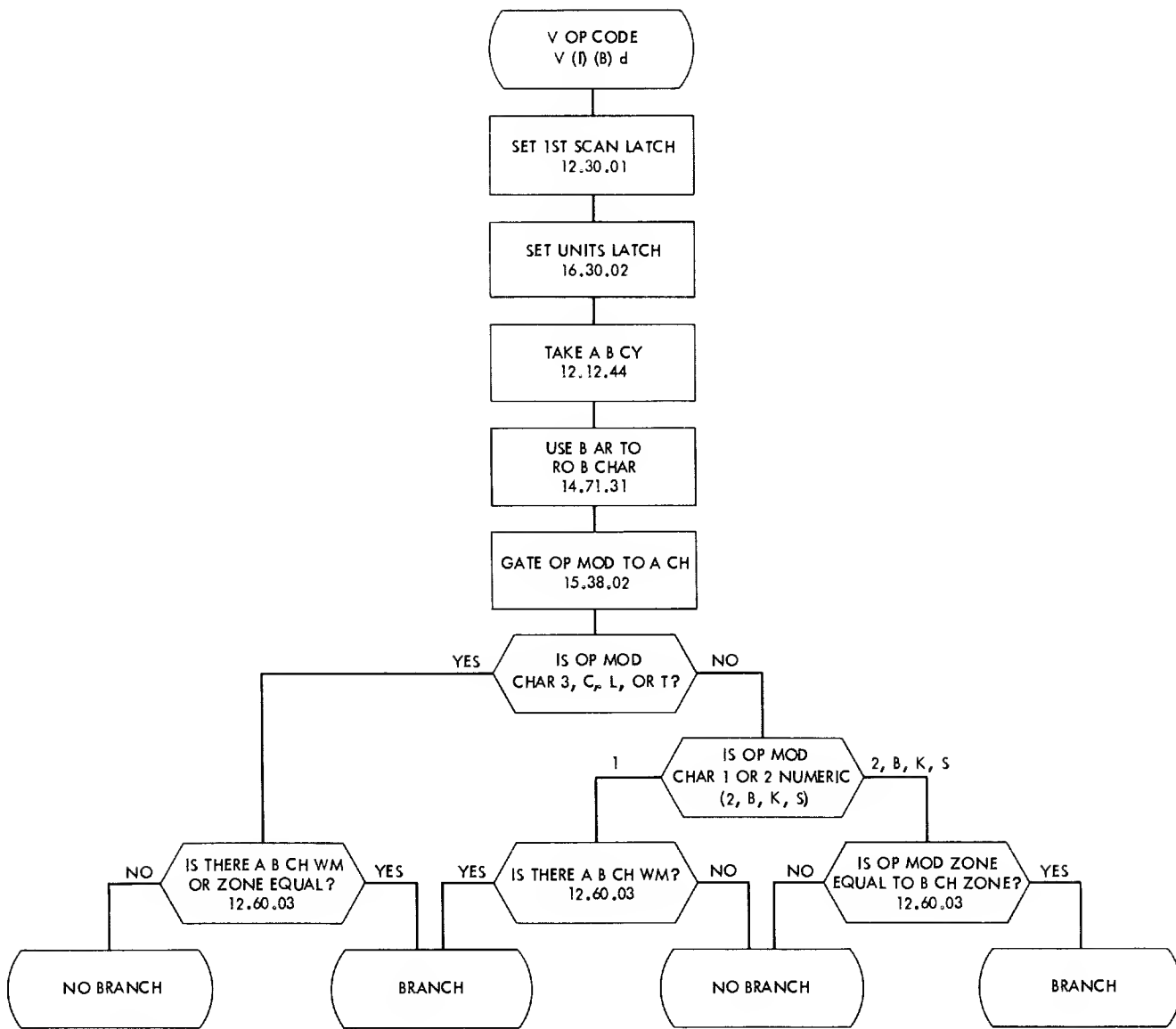


Figure 8.6-4 Branch If WM and or Zane



#### 8.6.04 Branch, if Bit Equal W ( I ) (B) d

##### Op-Code Function

This instruction compares the character at the B-address, bit by bit, with the d-character. If any bit in the character at the B-address matches any bit in the configuration of the d-character, the program branches to the I-address. The d-character can contain any character or any combination of bits that can exist in a single position of core storage. WM and C-bits are not compared.

##### Operation (Figure 8.6-3)

At I-ring-11 time, during instruction read-out, the Op-modifier register sets with the Op-modifier character. At I-ring-12 time, the 1st scan-control latch sets and a B-cycle starts. The character at the B-field address reads to the B-channel. The Op mod register is gated to the A-channel and the bits compare. Any bit on both channels starts a branch.

#### 8.6.05 Branch, if WM and/or Zone V ( I ) (B) d

##### Op-Code Function

This Op code tests the character that is located in the B-address for the condition specified by the d-character, and branches to the I-address, if the condition is met. The d-modifiers and the conditions they specify are as follows:

<u>d-Character</u>	<u>Bits</u>	<u>Control</u>	<u>Logic</u>
1	1	Word mark	13. 12. 17
2	2	No zone (No A, No B-bit)	13. 12. 17
B	B A 2	12-zone (A • B bits)	13. 12. 11
K	B 2	11-zone (B, A-bit)	13. 12. 16
S	A 2	Zero-zone (A, no B-bit)	13. 12. 10
3	B 21	Either a WM or no-zone	13. 12. 17
C	B A 21	Either a WM or 12-zone	13. 12. 15
L	B 21	Either a WM or 11-zone	13. 12. 16
T	A 21	Either a WM or zero-zone	13. 12. 10

##### Operation (Figure 8.6-4)

This operation duplicates the B-Op code until the character designated by the BAR reads to the B-channel. At this point, the bits on the B-channel are tested for the configuration specified by the d-modifier. The d-modifier was gated to the A-channel.

If the conditions imposed by the d-modifier (see the table in the preceding paragraph) are met, a branch starts. Otherwise, the program proceeds with the NSI.

The test of the B-address character is made by switching the result of zone bit compare (= or ≠) with the Op modifier 1- and/or 2-bits. The choice of d-characters makes it possible to test for several combinations of WM and zone bits.

## 8.7.00 MISCELLANEOUS

### 8.7.01 Store A-Address Register

#### Op-Code Function

The 1401 Q-Op code stores the contents of the A-address register from the previous operation in the three-position field that has its units position defined by the A-address of the store-A-address-register instruction. Word marks in the A-field are retained. The instruction format is: Q (AAA).

#### Operation (Figure 8.7-1)

The A-address of the Q-Op code reads into the AAR on the instruction read-out cycle. However, the AAR contains the address that must be stored. Therefore, the A-address left from the previous operation must move to the BAR before the new A-address reads in. This is done during a B-cycle that occurs just after the Q-Op is detected at I-Op time. After the B-cycle, the I-ring then advances, and the (AAA) address reads into the AAR.

The BAR contains a five-digit numerical 1410 address. In 1401 mode, a three-digit address with appropriate zones over the units and hundreds position must be stored. The units, tens and hundreds position of the BAR are stored on A-cycles during following steps of the A-ring. The zones over the units and hundreds that represent thousands- and ten-thousands-position values must be stored at the same time as the numerical portion of the character. To do this, the thousands- and ten-thousands positions of the BAR read out to the binary register during two C-cycles before the storage operation (Figure 8.7-2).

During the storage operation, at A-ring-2 time, the units position of the BAR reads out via the address exit channel to the A-data register, to the A-channel, and to assembly. At the same time, the A4 and A8 binary registers are gated out to the zone adder. The zone adder output is gated to the assembly area to combine with the numerical bits from the A-channel. The resulting character reads into storage in the location addressed by the AAR.

At A-ring-3 time, the tens position numerical bit reads out of the BAR, via the A-channel, to storage. No zones are involved. At A-ring-4 time, the hundreds-position numerical bit reads out of the BAR to assembly the same as the units position. The binary-register A1 and A2 outputs are gated through the zone adder to assembly to form the zone portion of the character to be stored.

### 8.7.02 Store B-Address Register

#### Op-Code Function

The 1401 H-Op code stores the contents of the B-address register from the previous operation in the three-position field that has its units position defined by the A-address of the store B-address register instruction. Word marks in the A-field are retained. The instruction format is: H (AAA).

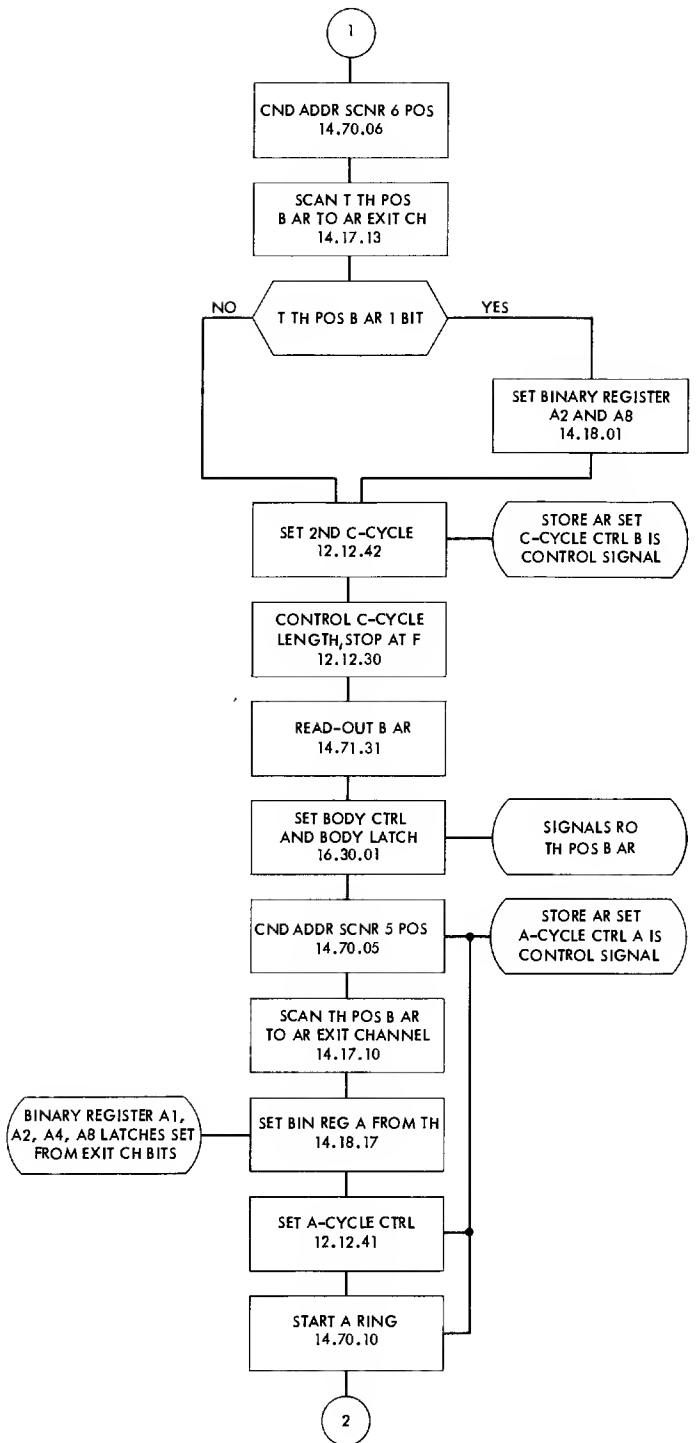
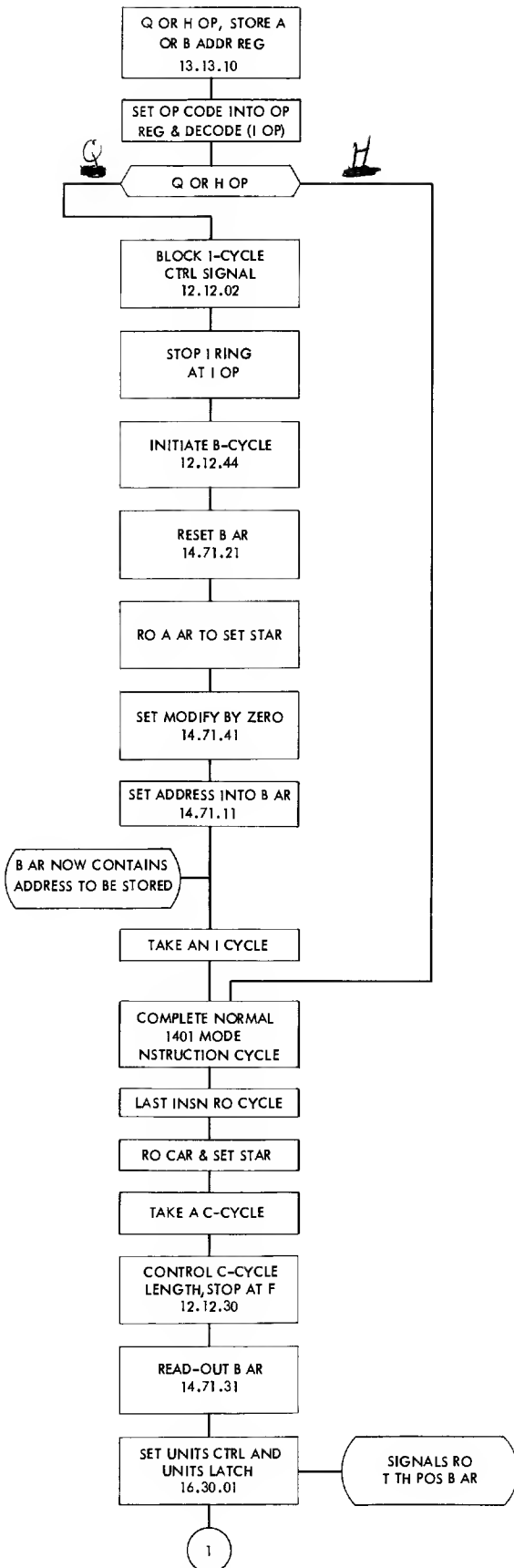
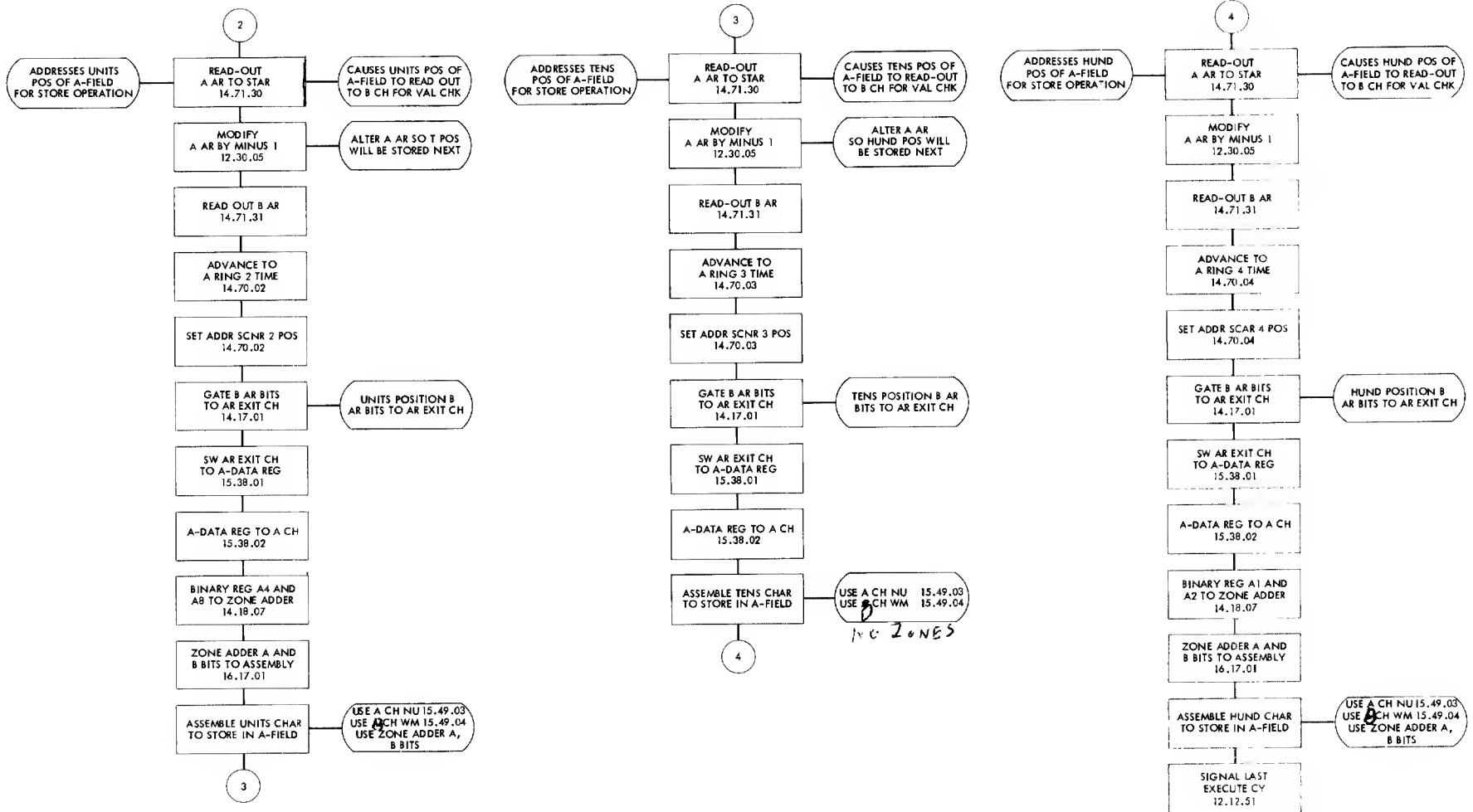
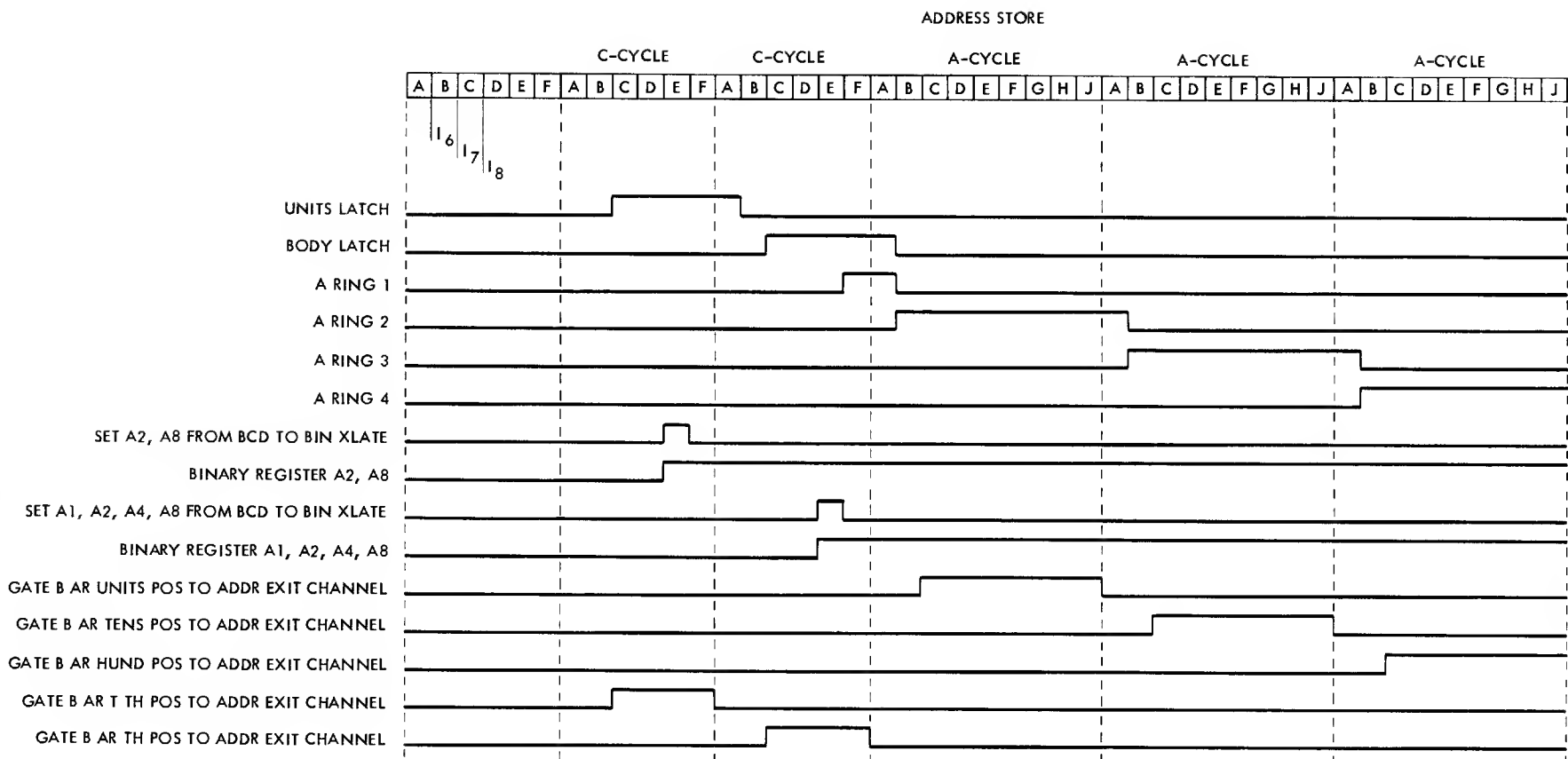


Figure 8.7-1A IBM 1401 Mode Store A or BAR

Figure 8.7-18 IBM 1401 Mode Store A or BAR



361



## Operation (Figure 8.7-1)

The (AAA) address reads into the AAR in the usual manner. Because the address to be stored is already in the BAR, no transfer from the AAR to the BAR is needed. Thus, the B-cycle after I-Op time is not necessary. The operation proceeds in the same manner as the Q-Op code.

### 8.7.03 Modify Address

#### Op-Code Function

The # (AAA) (BBB) instruction adds the three-character field specified by the A-address to the B-address. The three numerical positions and zones over the units and hundreds positions of each field added, and the three-position result (including appropriate zones) is stored in the B-field. Word marks are ignored and remain unchanged in both fields.

#### Operation (Figure 8.7-3)

Three sets of A- and B-cycles add the address from the B-field to the modifier in the A-field. The modified address is then stored back in the B-field. The zone bits over the B-field units and hundreds position must be added to the zone bits over the modifier. The problem is illustrated in the following example:

	A = 1
	B = 2   B = 8
Contents of B-address to be modified	9 6 5
	B = 2
Contents of A-address - modifier	<u>0 0 0</u>
	A = 1   A = 4
	B = 8
Sum	9 6 5
In 1410 numerical form	13965
	+ <u>2000</u>
	15965

The following sequence of events takes place (Figure 8.7-4)

1. First add cycle. The units position 5- and 0-digits add in the 1410 adder. The result switches to the assembly channel. The B-zone bit on the B-channel adds to no-bit in the zone adder. The resulting B-bit is gated to the zone assembly by a forced adder no-carry.
2. Second add cycle. The tens-position 6- and 0-digits add in the 1410 adder. The result switches to the assembly channel. Any zone bits (index tags) over the 6 in the B-field are gated from the B-channel to assembly. No zone adder operation is involved.

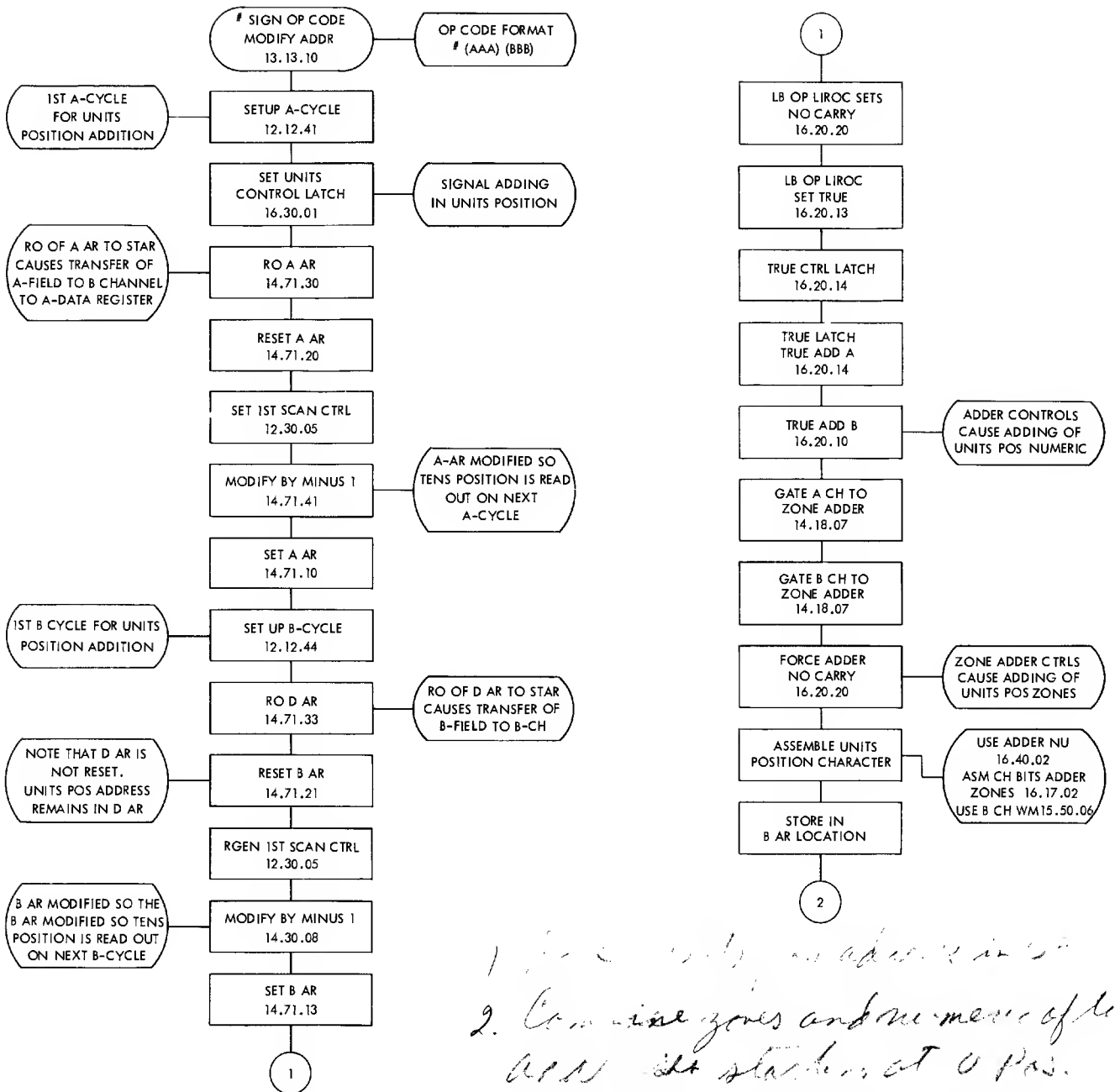


Figure 8.7-3A IBM 1401 Mode Modify Address

1. [unclear] address in [unclear]
2. Combine zones and numerics of two  
addr. start at 0 pos.
3. At 4 Pos. convert address to  
to zones. If a zone cannot be converted  
to a zone over unit 12.

Figure 8.7-38 IBM 1401 Mode Modify Address

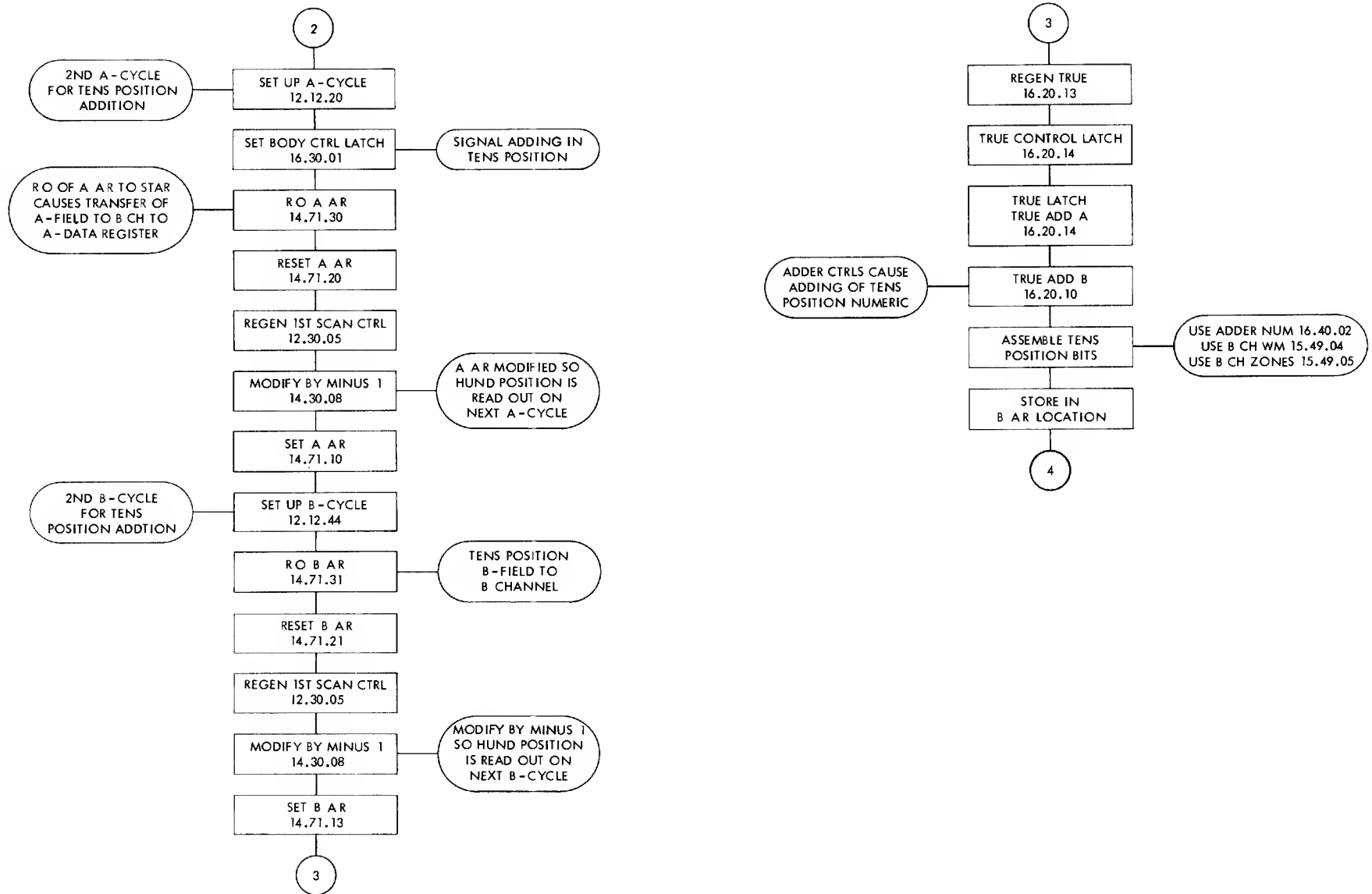
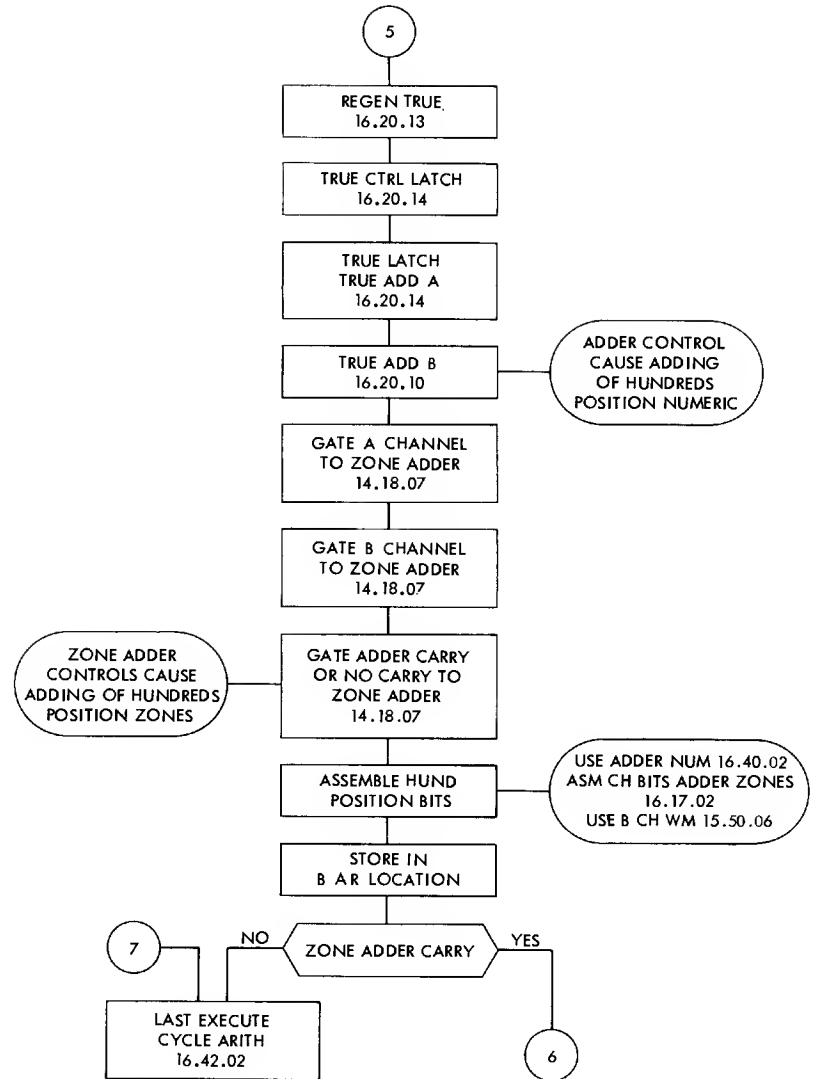
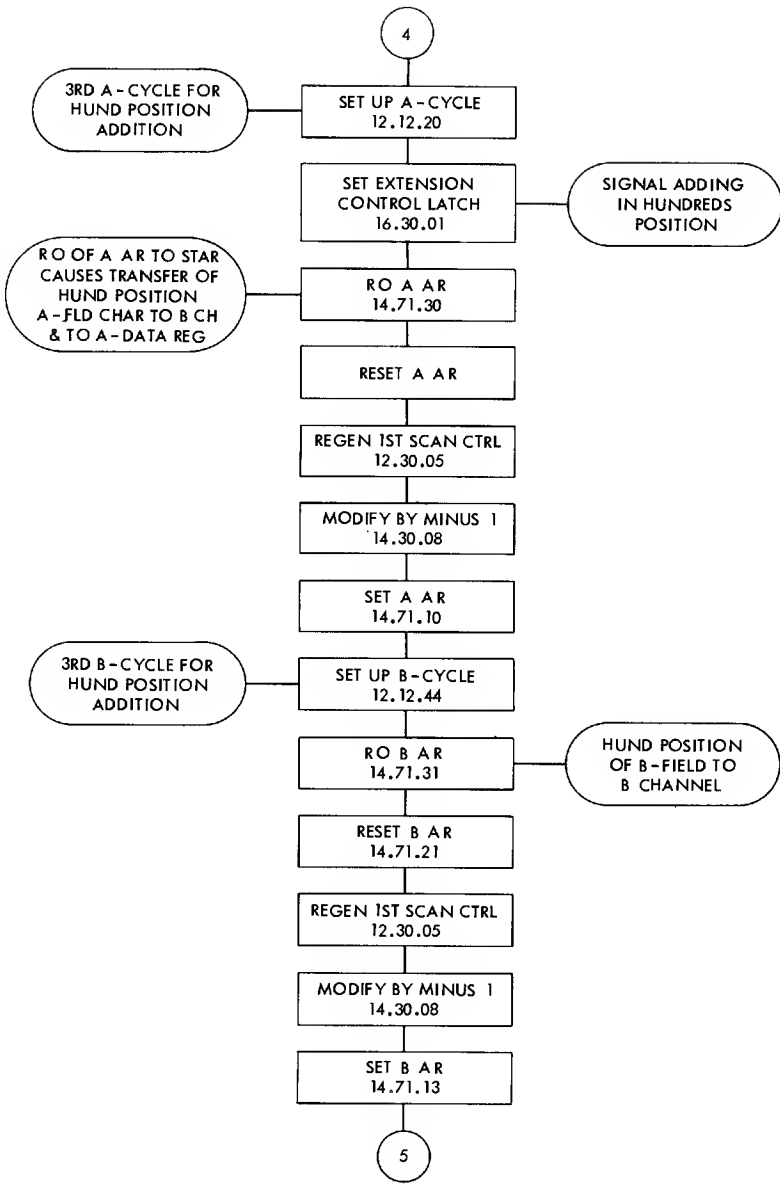




Figure 8.7-3C IBM 1401 Mode Modify Address



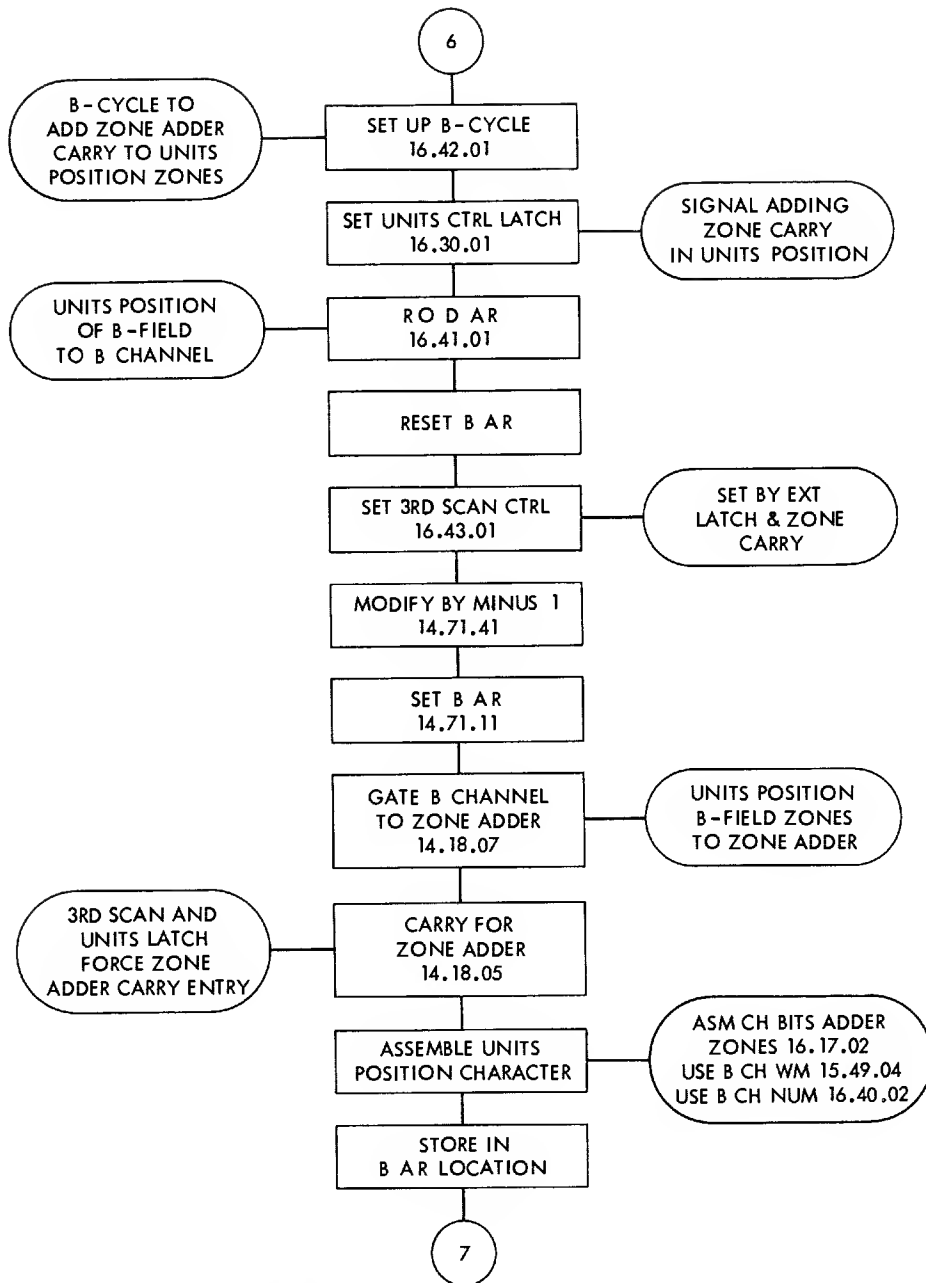


Figure 8.7-3D IBM 1401 Mode Modify Address

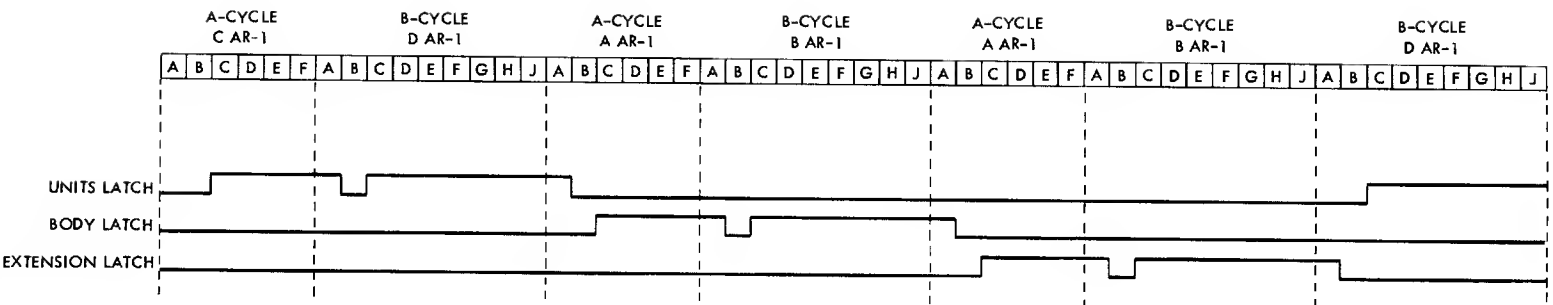


Figure 8.7-4 IBM 1401 Mode Modify  
Address Sequence

3. Third add cycle. The hundreds position 9- and 0-digits add. The result switches to the assembly channel. The A- and B-zone bits on the B-channel add to the B-zone bit on the A-channel in the zone adder. The resulting A-bit is gated to the zone assembly by the adder no-carry. In addition, a zone-adder-carry results. This must be added to the zones in the units position. A B-cycle is taken to do this.
4. Special B-cycle, caused by zone-adder-carry. On the first add cycle (B-cycle portion), the DAR reads out to address storage, but does not reset. Instead, the BAR resets and is modified, and the DAR remains unchanged, with the units position B-address intact. This address is used on the special B-cycle to read out the units position of the B-field to the B-channel. The previously stored B-bit then adds to the zone-adder-carry to result in A- and B-bits. These are gated to the assembly channel along with the B-channel numerical bits.
5. Signal last-execute cycle. The last-execute cycle signal develops after the third add cycle, if no zone-adder-carry occurs. If a zone-adder carry takes place, the last-execute cycle signals after the special B-cycle caused by the zone-adder-carry.

The units, body and extension latches signal which position is added.

<u>Latch</u>	<u>Position Added</u>
Units	Units position
Body	Tens position
Extension	Hundreds position

#### 8.7.04 No Operation (N)

##### Op-Code Function

The N-Op code performs no operation. It can substitute for the Op code of any instruction to make that instruction ineffective.

##### Operation (Figure 8.2-1)

When the Op registers detect a No-Op Op code, they allow the I-ring to advance and I-cycles to be taken until the next Op code, as defined by a word mark, reads out of storage. The word mark conditions Last Insn RO Cycle, Last I-Cycle, and Last Execute Cycle to start the next instruction read-out operation.

#### 8.7.05 Halt ( . )

##### Op-Code Function

The ( . ) Op code stops the machine and starts a stop print-out operation. Pressing the start key starts the program at the next instruction in sequence.

##### Operation (Figure 8.7-5)

The last instruction read-out cycle (determined by the presence of a B-channel WM

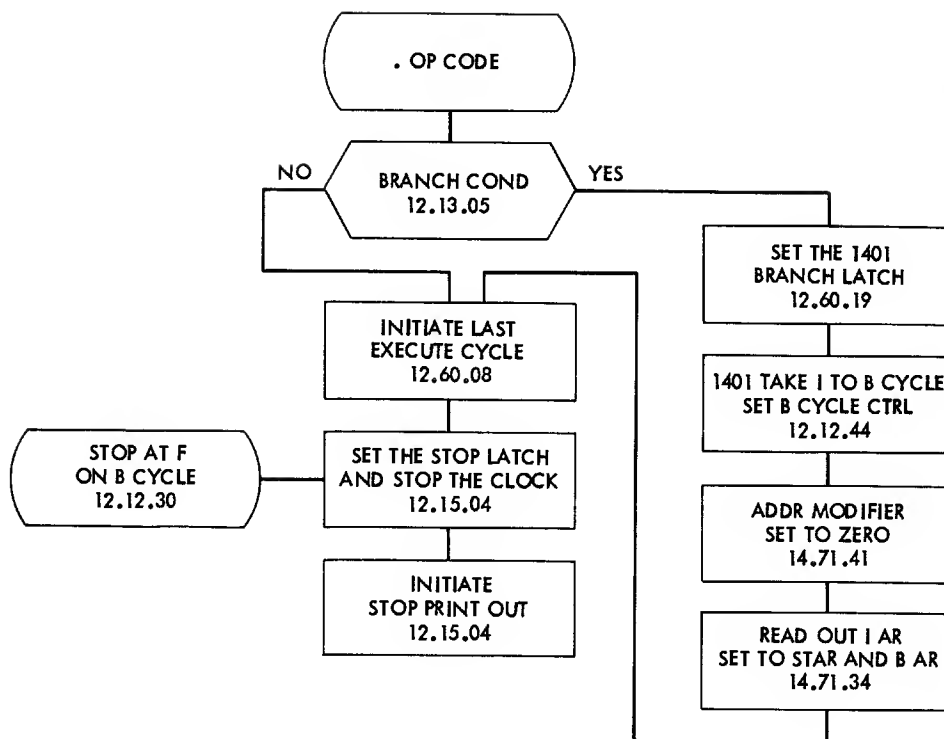


Figure 8.7-5 Halt, and Halt and Branch

bit at I-ring-1 time) causes a simultaneous last-execute cycle. This sets the stop latch, stops the logic clock, and starts a stop print-out operation.

#### 8.7.06 Halt and Branch Op . (I)

Operation (Figure 8.7-5)

The I-address reads into the AAR. A B-cycle then starts. During the B-cycle the IAR reads out to the STAR, and through the modifier (set to modify-by-zero condition) to the BAR. The last-execute cycle stops the logic clock and starts the stop print-out.

When the start key is pressed, the 1401 branch latch sets to read out the AAR to STAR for the I-Op cycle of instruction read-out.

#### 8.7.07 Set Word Mark , (A) (B)

Op-Code Function

The , Op code causes a word mark to set at each address specified in the instruction. The data at each address is undisturbed. If this instruction is given with one address (A-address), a word mark is set at the A-address only. If the address instruction is indexed, a word mark sets at the locations specified by the indexed A-address. However, if this instruction is given with no A-address (no-address instruction), word marks are set at the locations specified by the addresses in both the A- and B-address registers (contents from previous operation).

Operation

One A- and One B-cycle are required to execute this operation. During A-cycle, the A-address character reads from storage to the B-channel. It is then gated through the assembly and back into storage, accompanied by a WM bit. Also at this point, a check bit is added or removed to maintain parity.

During B-cycle the character at the B-address reads from storage and is treated similarly. If only one address is specified, the A-data address is stored in both the A- and B-address registers during instruction RO. Therefore, when the execute phase begins for a single address instruction the A-data address is in both the AAR and BAR. Thus a WM is set in the same location twice, once during A-cycle, once during B-cycle.

If this instruction is given with no address, WM's are set in A- and B-address locations specified in the previous instruction. The operation is the same as a two-address instruction.

#### 8.7.08 Clear Word Mark □ (A) (B)

Op-Code Function

The Op code causes word marks cleared from the locations specified by the A- and B-addresses of the instruction. The data at each address is undisturbed. If

this instruction is given with one address, a word mark clears at the A-address only. If the instruction is indexed, a word mark clears at the location specified by the indexed address. However, if this instruction is given with no addresses, word marks clear at the locations specified by the addresses in both the A- and B-address registers from the previous operation.

#### Operation

This operation is executed in much the same manner as set word mark (section 8.7.07 Set Word Mark). One A- and one B-cycle are taken. The character to be treated is handled in the same manner. The same Op-code grouping lines are activated. The significant difference is that the WM bit is removed (instead of being set) as the character is gated through assembly to storage.

This is accomplished by controlling the assembly to Use No WM.

#### 8.7.09 Clear Storage / (A)

##### Op-Code Function

The / Op code causes as many as 100 positions of core storage to be cleared of data and word marks. Clearing starts at the A-address and is forward-scanned to the nearest hundreds position. The cleared area sets to blanks.

##### Operation (Figure 8.7-6)

The operation is executed by a series of B-cycles during which the assembly is controlled to Use No WM's, Zones or Numerics. Clearing starts with the location specified by the A-address, and continues to the nearest even hundreds position. The Mod by Minus-One condition that is still ON at logic gate D-time recognizes the even-hundreds address. This indicates a borrow-one-from-hundreds position. It sets the even-hundreds latch, and ends the execute phase.

When this instruction is given without an address, the contents of the B-address register from the previous operation are used as the A-address.

#### 8.7.10 Clear Storage and Branch / (I) (B)

##### Op-Code Function

This instruction performs the same operation as / (A) except that the clearing starts at the B-address. The I-address specifies the location of the next instruction. This is an unconditional branch.

##### Operation (Figure 8.7-6)

As described for the clear storage operation, Clear Storage and Branch is executed by a series of B-cycles. Because the last instruction RO cycle occurs at I-ring-11, the 1401 branch latch is set ON. When the even-hundreds address sets ON and is gated with the 1401 branch latch ON, No Scan Control sets and another B-cycle is taken to store the address in the IAR. The next instruction reads out, starting from the address specified by the AAR.

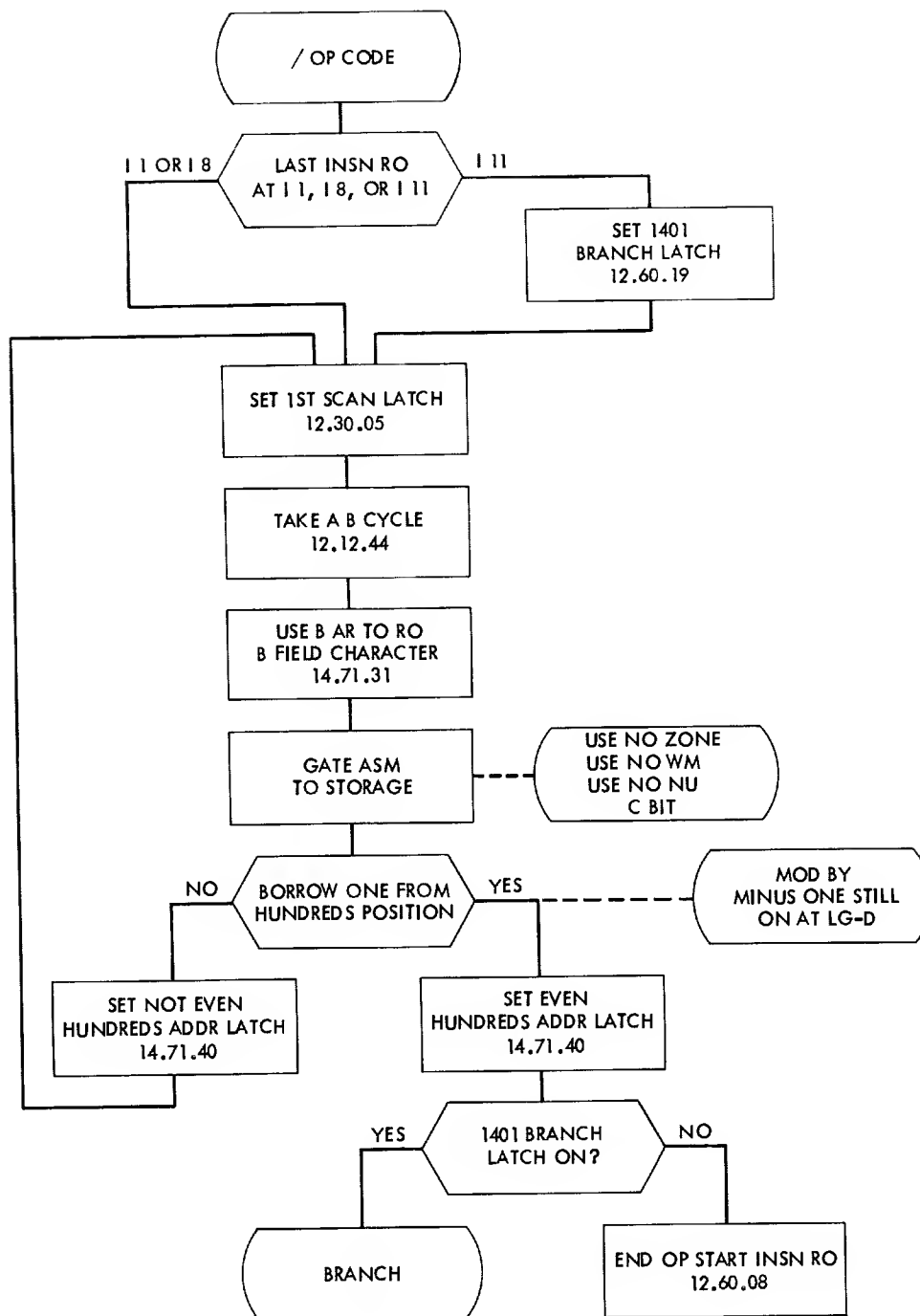


Figure 8.7-6 Clear Op Code



## 8.8.00 INPUT-OUTPUT INSTRUCTIONS

Four major differences arise when processing IBM 1401 I/O instructions on the IBM 1410:

1. Specific input-output areas are assigned in the CPU core-storage unit. Cards read into storage locations 00001 through 00080. Cards are punched from storage locations 00101 through 00180. Information prints from storage locations 00201 through 00332.
2. A 1401 instruction can be given to address more than one I/O unit. For example, the 1401 Op code 7 causes the printer to print a line, the card reader to read a card, and the card punch to punch a card (in that order).
3. A 1401 I/O instruction can include an I-address to cause an unconditional branch after the I/O operation is completed.
4. If an I/O error occurs, or if an addressed I/O unit is not ready or is busy, the CPU stops and waits either until the operator corrects the not-ready or error condition, or until the addressed unit becomes not busy. This means that the 1410 I/O channel status indicators are never set when operating in the 1401 mode.

### 8.8.01 Read a Card

#### Op-Code Function (1)

The 1401 read-a-card Op code 1 causes all 80 columns of information to read into core-storage locations 00001 to 00080.

#### Operation (Figures 8.8-1 and 8.8-2)

A reader operation in 1401 mode causes the contents of the 80-position read-storage unit in the synchronizer to transfer to the core-storage unit in the CPU. The starting address, 00001 is generated in the CPU. 1401 instructions cause the 1410 to operate in the move mode; there is no translation of word-separator characters. Word marks that previously were set in 00001 through 00080 read out on the B-channel, combine in assembly with the characters from read storage, and read back into the CPU core-storage unit.

The 1 Op code brings up the 1401 card-print Ops signal to indicate that one or more of the input-output operations is to be performed. At last instruction RO cycle time, a 4-position I/O scan-ring scans the Op register 2 (print), Op register 1 (read) and Op register 4 (punch) lines in sequence. If the tested Op register output is not up, the ring advances at the next 1st-clock-pulse time to test the next Op register output in sequence. Since the Op register 1 line is up, the scan-ring advance stops, and the read trigger (13.70.01) remains ON until the record from read storage transfers into CPU core storage.

The E-channel is selected automatically by the 1401 read operation. Because the 1401 does not have status indicators, E-channel status indicators are not set before starting transmission. If the reader is not ready or busy, the CPU stops,

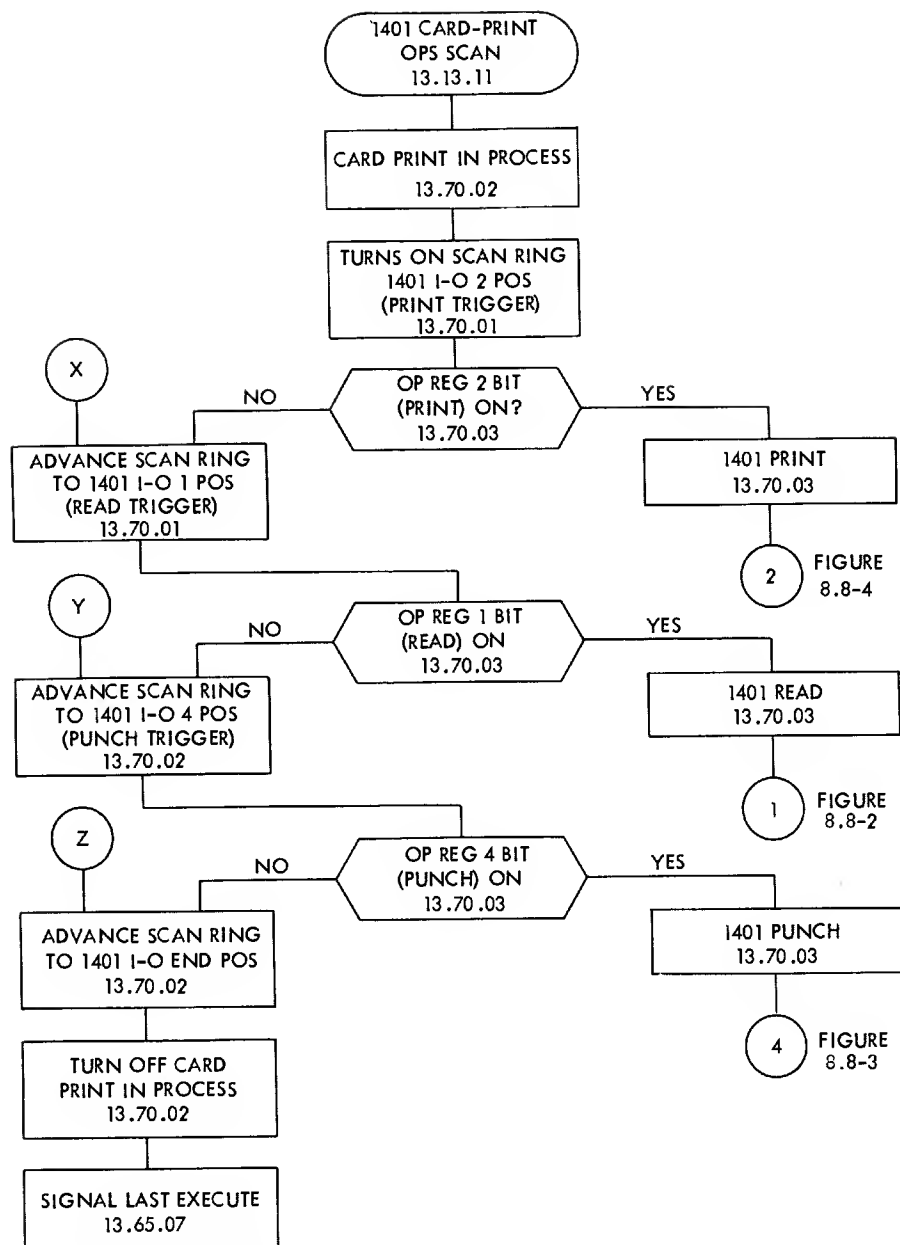


Figure 8.8-1 IBM 1401 Card-Print Ops Scan

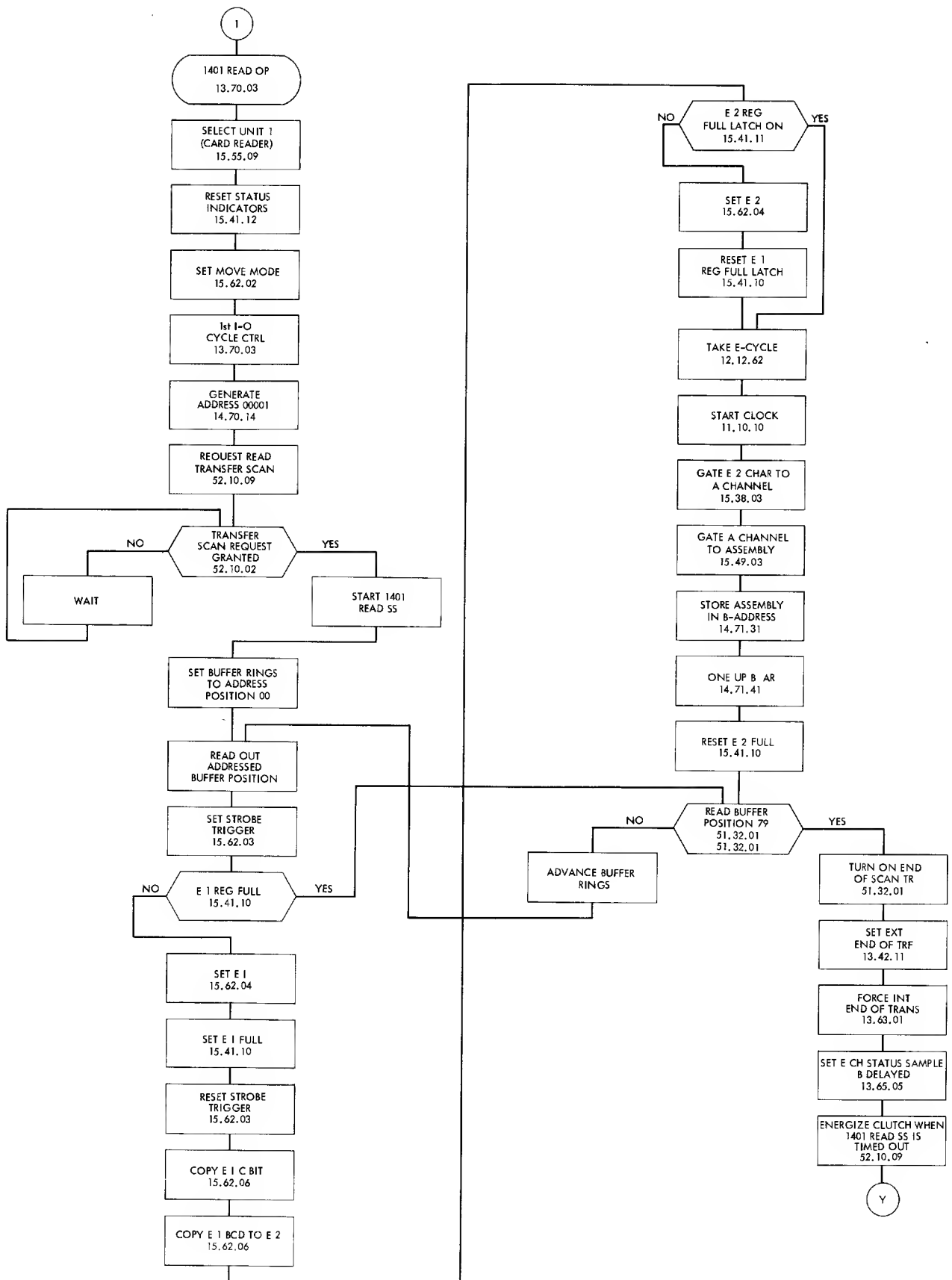


Figure 8.8-2 IBM 1401 Mode Read Op

and must be restarted manually. The E-channel unoverlap-in-process latch sets so the transmission takes place via the E-channel in the unoverlap mode. The address 00001 is generated and gated to the BAR. The transfer of data takes place in the usual manner, and the BAR is updated on each B-cycle. After the transfer, the next card reads into reader storage.

When the buffer end-of-transfer signal returns to the CPU, the E-channel status sample B-delay comes ON. This, in turn, advances the scanning to the end position provided that no punch operation is indicated. Input-output controls turn OFF, and a last-execute cycle signals.

## Circuits

### 1. Recognize the card-print operation.

<u>Signal</u>	<u>Control</u>	<u>Logic</u>
1401 card print ops	(not) Ctrl reg disable 1401 mode Op crdr, not A, not B, not 8	13. 13. 11

### 2. Scan for print or read or punch.

<u>Signal</u>	<u>Control</u>	<u>Logic</u>
1401 print trigger	1401 card print ops Last insn RO cycle, LG-E	13. 70. 01
Not 1401 card or prtr mode	(not) Op reg 4-bit (not) Read trigger (not) Punch trigger	13. 70. 03

Because this is not a print operation (Op Reg 2-Bit is not ON), the scan ring advances to the 1401 I/O 1-position (read trigger).

<u>Signal</u>	<u>Control</u>	<u>Logic</u>
1401 I/O ring advance	(not) 1401 card or prtr mode	13. 70. 01
1401 read trigger	1401 I/O ring advance 1401 print trigger	13. 70. 01
1401 Read	1401 Read trigger Op Reg 1 Bit	13. 70. 03

The 1401 read signal prevents the development of the Not 1401 Card or Prtr Mode signal. Thus, the scanning does not advance and the read trigger remains ON until read storage transfer-scans to core storage.

### 3. Signal the card print-in-process (i. e. , read).

<u>Signal</u>	<u>Control</u>	<u>Logic</u>
1401 set cd print-in-proc	1401 card or print op code Last insn RO cycle, LG-E	13. 70. 01
1401 card print-in-proc	1401 set cd print-in-proc	13. 70. 02

The 1401 card print-in-process trigger remains ON until the scan ring advances to the 1401 I/O end position. On a read-only operation, this occurs immediately after read storage empties because no punch operation takes place.

4. Select E-channel and unit 1.

<u>Signal</u>	<u>Control</u>	<u>Logic</u>
E-ch select unit 1	1401 read	13. 50. 03
Unit 1 select to I/O	1401 read	13. 50. 03
E-ch select any buffer	E-ch select unit 1	13. 60. 03
E-ch in mode (latch)	1401 read trigger	15. 62. 01
E-ch input mode	E-ch in mode	15. 62. 01
E-ch input op to buffer	E-ch in mode	15. 62. 01
Gate I/O sync to E1	E-ch sel any buffer E-ch input mode	15. 62. 07

5. Set up the E-channel for processing.

<u>Signal</u>	<u>Control</u>	<u>Logic</u>
Card print reset (trigger)	1401 read, 2nd clock pulse clamped	13. 70. 03
1st I/O cycle control	Card print reset (trigger) 2nd clock pulse clamped	13. 70. 03
E-ch unovlp-in-process	1401 card print-in-process 1st I/O cycle control	13. 60. 04
E-ch-in-process	E-ch unovlp-in-process	13. 60. 04

6. Generate address 00001.

<u>Signal</u>	<u>Control</u>	<u>Logic</u>
RO 00001 index addr	1st I/O cycle control 1401 read trigger	14. 70. 14
RO fixed addr	RO 00001 index addr	14. 70. 14
AR bus (00001)	RO fixed addr and RO 00001 index addr	14. 15. 01 to 14. 15. 12

7. Request a read transfer scan.

<u>Signal</u>	<u>Control</u>	<u>Logic</u>
Ready-to-buffer	E-ch-in-process E-ch select any buffer E-ch input mode	13. 70. 04
Ready	Ready-to-buffer	51. 40. 04
Rd trans req	Ready Rd ready (not) Reader busy Select unit 1	52. 10. 09
Rd priority gate	Rd trans req	52. 10. 07
Rd pr req trigger (on)	Rd priority gate Time 080-090	52. 10. 01
Gate on rd scan tr	Rd pr req tr (not) Same scan (not) PT scan req	52. 10. 01
Rd scan	Gate on rd scan tr Time 030-040	52. 10. 02

#### 8. Read out the addressed buffer position.

<u>Signal</u>	<u>Control</u>	<u>Logic</u>
TimePl 1 latch (on)	Read scan (not) Single cycle mode	51. 30. 02
Time Pl 2 latch (on)	Time Pl 1 latch (on) Time 000-010	51. 30. 02
Timing pulse gate	Time Pl 2 latch (on)	51. 30. 02
Rd pulse L latch (on)	Time 000-010 Not clock 2	51. 30. 05
Rd pulse L latch (off)	Time 040-050	51. 30. 05
Rd pulse S latch (on)	Time 010-020	51. 30. 05
Rd pulse S latch (off)	Time 040-050	51. 30. 05
Read pulse 1 010-040	Rd pulse S latch Timing pulse gate	51. 30. 05
Read pulse 2 005-040	Rd pulse L latch Timing pulse gate	51. 30. 05

#### 9. Set the strobe trigger.

<u>Signal</u>	<u>Control</u>	<u>Logic</u>
Trans scan	Rd tfr scan	51. 40. 40
Strobe latch	Trans scan Time 020-030	51. 40. 43

<u>Signal</u>	<u>Control</u>	<u>Logic</u>
I/O to CPU trans	Rd trf scan	51. 40. 40
Buffer strobe	I/O to CPU trans Time 080-090 Strobe latch	51. 40. 43
E-ch strobe trigger	Buffer strobe E-ch select any buffer	15. 62. 03

10. Set the E1 register, set the E1 full latch, and reset the strobe trigger.

<u>Signal</u>	<u>Control</u>	<u>Logic</u>
Set E1 reg	E-ch strobe trigger E-ch input mode (not) E1 reg full	15. 62. 04
E-ch strobe trigger (off)	E-ch input mode Set E1 reg	15. 62. 03
E1 full	Set E1	15. 41. 10

11. Transfer E1 to E2.

<u>Signal</u>	<u>Control</u>	<u>Logic</u>
E-ch move mode	1401 card print-in-process	15. 62. 02
E-ch reset	Card print reset	15. 41. 12
Copy E1 BCD to E2	(not) E1 reg word separator	15. 62. 06
E2 reg full latch (off)	E-ch reset	15. 41. 10
Set E2 reg	E1 reg full latch (on) E2 reg full latch (off)	15. 62. 04
E2 reg full latch (on)	Set E2 reg (not) E1 reg word separator	15. 41. 10
E1 reg full latch (off)	E-ch input mode Set E2 reg	15. 41. 10

12. Signal that an E-cycle is required.

<u>Signal</u>	<u>Control</u>	<u>Logic</u>
E-cycle required	E-ch input mode E-ch-in-process E2 reg full	12. 12. 62
E-cycle ctrl	E-cycle required, 2nd clock pulse LG-Z, (not) F-cycle required	12. 12. 66
E-cycle	E-cycle ctrl, LG-B or 5	12. 12. 66

13. Control the clock.

<u>Signal</u>	<u>Control</u>	<u>Logic</u>
Comp disable cycle	E-ch unovlp-in-process	12. 12. 60

This holds the clock OFF, until E-cycle-required comes ON.

<u>Signal</u>	<u>Control</u>	<u>Logic</u>
Logic gate A (unclamped)	E-cycle required E-ch unovlp-in-process	11. 10. 10

14. Gate the E2 character to the A-channel.

<u>Signal</u>	<u>Control</u>	<u>Logic</u>
Gate E2 data reg to A-ch	E-ch input mode E-cycle	15. 38. 03

15. Gate A-channel to assembly.

<u>Signal</u>	<u>Control</u>	<u>Logic</u>
Use A-ch zones	In cy GM-WM ctrl A-ch valid or ast switch (off) Odd-parity cycle	15. 49. 03
Use A-ch nu	In cy GM-WM ctrl A-ch valid or ast switch (off) Odd-parity cycle	15. 49. 03
Use B-ch WM	Input cycle Move cycle	15. 49. 04

16. Store the assembly channel in the B-address.

<u>Signal</u>	<u>Control</u>	<u>Logic</u>
B or E or F-cycle ctrl	E-cycle ctrl	12. 12. 02
B or E or F-cycle	B-cycle	12. 12. 07
RO BAR	E-cycle ctrl E-ch unovlp-in-process	14. 71. 31
Reset BAR	B or E or F-cycle ctrl LG-Early B	14. 71. 21
Set BAR	B or E or F-cycle ctrl, LG-B or C B or E or F-cycle, LG-D or E or F	14. 71. 11
Addr mod set to plus one	E-cycle ctrl	14. 71. 41



17. Energize the read clutch.

<u>Signal</u>	<u>Control</u>	<u>Logic</u>
1401 read latch	Rd trans req 1401 mode	52. 10. 08
Rd feed gate (latch)	1401 read latch, End of SS timing	52. 10. 08
Read feed (trigger)	Rd feed gate, Proc feed rd	52. 10. 09
Rd clutch	Read feed	52. 10. 09

18. End the external transfer, and develop sample pulses.

<u>Signal</u>	<u>Control</u>	<u>Logic</u>
E-ch ext end-of-transfer	Buffer-end-of-transfer E-ch select any buffer 1st clock pulse	13. 42. 11
E-ch last input cycle	E-ch ext end-of-transfer (not) E1 reg full (not) E2 reg full E-ch cycle, E-ch input mode Logic gate C or T	13. 63. 02
E-ch int end-of-transfer	E-ch last input cycle LG-F or W	13. 63. 01
E-ch int end-of-trf delayed	E-ch last input cycle Logic gate Z	13. 63. 01
E-ch status sample B	E-ch int end-of-trf delayed E-ch ext end-of-transfer Logic gate Z, (not) 2nd clock pulse	13. 65. 05
E-ch second sample B	E-ch status sample B (not) 2nd clock pulse	13. 65. 05
E-ch status sample B delayed	E-ch second sample B (not) 2nd clock pulse	13. 65. 05

19. Advance the scan ring to the end.

<u>Signal</u>	<u>Control</u>	<u>Logic</u>
1401 I/O ring advance	E-ch status sample B delayed	13. 70. 01
1401 I/O 4-pos (punch)	1401 read trigger 1401 I/O ring advance	13. 70. 02
(not) 1401 card or prtr mode	(not) read, (not) punch, (not) print	13. 70. 03
1401 I/O ring advance	(not) 1401 card or prtr mode	13. 70. 01

<u>Signal</u>	<u>Control</u>	<u>Logic</u>
1401 I/O end pos	1401 punch trigger 1401 I/O ring advance	13. 70. 02
1401 card print-in-proc (off)	1401 I/O ring advance 1401 I/O end	13. 70. 02

## 20. Signal Last Execute cycle.

<u>Signal</u>	<u>Control</u>	<u>Logic</u>
Last execute cycle I/O	1401 I/O end (not) 1401 branch latch	13. 65. 07
Last execute cycle	Last execute cycle I/O	12. 12. 51

### 8. 8. 02 Punch a Card

#### Op-Code Function (4)

The 1401 punch-a-card Op code (4) punches the information in the 1410 core-storage locations 00101 to 00180 into a card. Word marks are not disturbed by the punch-out.

Operation (Figures 8. 8-1 and 8. 8-3)

The 4 Op code brings up 1401 card print Ops to allow the 1401 I/O scan ring to advance. The scan-ring print trigger turns ON, but since there is not an Op register 2-bit, the ring advances to the read-trigger position. There is no 1-bit in the Op register, so that the ring advances to the punch-trigger position. The 4-bit in the Op register combines with the punch trigger on output to signal a punch operation. A punch transfer scan is requested, and the transfer takes place over the E-channel in the unoverlapped, move mode. Eighty characters transfer to punch storage. When the transfer is complete, the punch is set in motion to punch the record, and the CPU is released for further processing.

#### Circuits

##### 1. Recognize the card-print operation.

<u>Signal</u>	<u>Control</u>	<u>Logic</u>
1401 card print opr	(not) ctrl reg disable 1401 mode Op dcdt not A, not B, not 8	13. 13. 11

##### 2. Scan for Print, Read, or Punch.

<u>Signal</u>	<u>Control</u>	<u>Logic</u>
1401 print trigger	1401 card print ops Last insn RO cycle Logic gate F	13. 70. 01

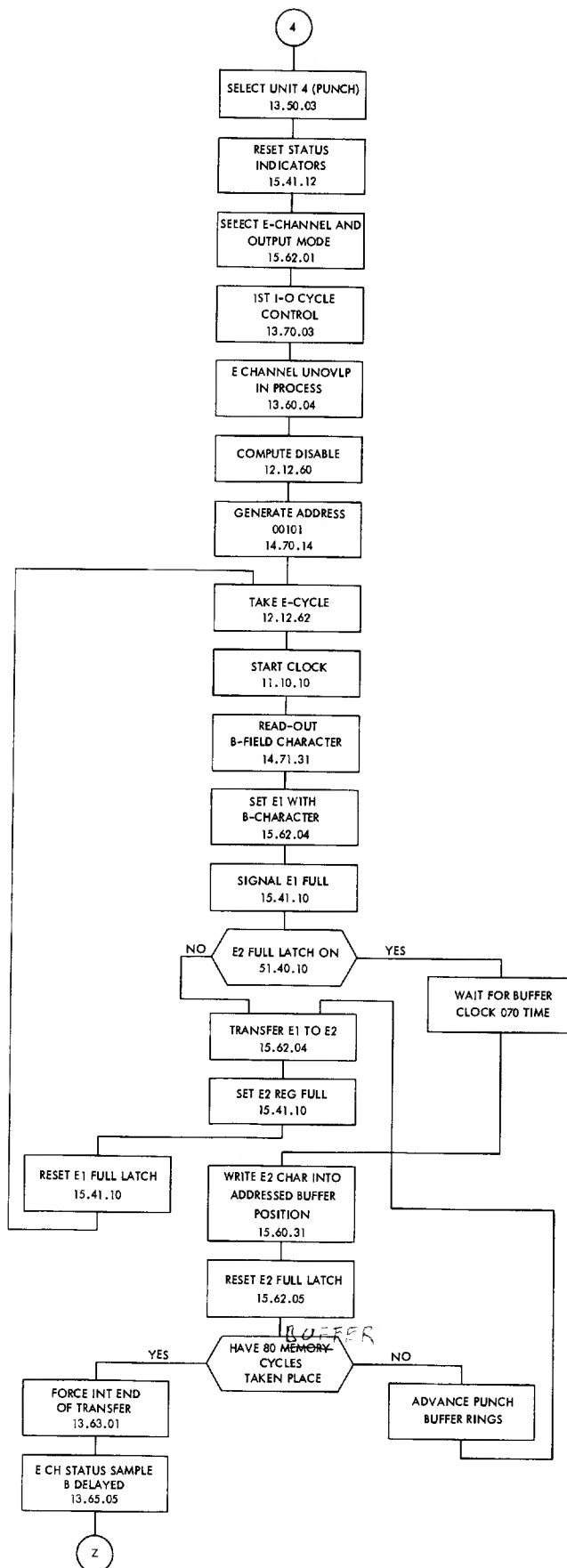


Figure 8.8-3 IBM 1401 Mode Punch Op

<u>Signal</u>	<u>Control</u>	<u>Logic</u>
Not 1401 card or prtr mode	(not) 1401 punch (not) 1401 read (not) 1401 print	13. 70. 03
1401 I/O ring advance	(not) 1401 card or prtr mode	13. 70. 01

Because this is not a print operation (Op register 2-bit is not ON), the scan ring advances to the 1401 I/O 1-position (read trigger). Because this is also not a 1401 read operation (not Op register 1-bit), the ring advances again to the 1401 I/O 4-position (punch trigger).

<u>Signal</u>	<u>Control</u>	<u>Logic</u>
1401 punch	1401 punch trigger Op reg 4-bit	13. 70. 03

The 1401 punch signal prevents the development of the Not 1401 Card or Prtr Mode signal. The 1401 I/O ring advance signal is not developed, so that the punch trigger remains ON until the punch transfer is complete.

### 3. Signal Card Print-in-Process.

<u>Signal</u>	<u>Control</u>	<u>Logic</u>
1401 set cd print-in-proc	1401 card or print op code Last insn RO cycle, LG-E	13. 70. 01
1401 card print-in-proc	1401 set cd print-in-proc	13. 70. 02

The 1401 card print-in-process trigger remains ON until the punch buffer is filled.

### 4. Select E-channel and Unit 4.

<u>Signal</u>	<u>Control</u>	<u>Logic</u>
E-ch select unit 4	1401 punch	13. 50. 03
Unit 4 select to I/O	E-ch select unit 4	13. 50. 03
E-ch select any buffer	E-ch select unit 4	13. 60. 03
E-ch out mode (trigger)	1401 punch trigger	15. 62. 01
E-ch output op to buffer	E-ch out mode	15. 62. 01
E-ch output mode	E-ch out mode	15. 62. 01
Gate asm ch to E1 input	E-ch output mode (not) Control reg disable	15. 62. 07

### 5. Set up the E-channel for processing.

<u>Signal</u>	<u>Control</u>	<u>Logic</u>
Card print reset (trigger)	1401 punch 2nd clock pulse clamped	13. 70. 03

<u>Signal</u>	<u>Control</u>	<u>Logic</u>
1st I/O cycle control	Card print reset 2nd clock pulse clamped	13. 70. 03
E-ch unovlp-in-process	1401 card print-in-process 1st I/O cycle control	13. 60. 04
E-ch-in-process	E-ch unovlp-in-process	13. 60. 04

6. Generate address 00101.

<u>Signal</u>	<u>Control</u>	<u>Logic</u>
RO 00101 index addr	1st I/O cycle control 1401 punch trigger	14. 70. 14
RO-fixed addr	RO 00101 index addr	14. 70. 14
AR bus (00101)	RO fixed addr and RO 00101 index addr to	14. 15. 01 14. 15. 12

7. Request an E-cycle.

<u>Signal</u>	<u>Control</u>	<u>Logic</u>
E-cycle required	E-ch output mode (not) E-ch int end-of-transfer E1 reg full latch (off) (not) E-cycle, any last gate	12. 12. 62
(Start logic clock)	E-cycle-required E-ch unovlp-in-process	11. 10. 10
E-cycle ctrl	E-cycle required 2nd clock pulse Logic gate Z (not) E-ch int end-of-transfer (not) F-cycle required	12. 12. 66
E-cycle	E-cycle ctrl Logic gate B or S	12. 12. 66

8. Read out the B-address to the B-channel.

<u>Signal</u>	<u>Control</u>	<u>Logic</u>
B or E or F-cycle ctrl	E-cycle ctrl	12. 12. 02
B or E or F-cycle	B-cycle	12. 12. 07
RO BAR	E-cycle ctrl E-ch unovlp-in-process	14. 71. 31
Reset BAR	B or E or F-cycle ctrl LG-early B	14. 71. 21
Set BAR	B or E or F-cycle ctrl Logic gate B or C	14. 71. 11
Addr mod set to plus one	E-cycle ctrl	14. 71. 41

9. Gate the B-channel character to the E1 input.

<u>Signal</u>	<u>Control</u>	<u>Logic</u>
Output cycle	E-ch output mode, E-cycle	13. 60. 06
Output field cycle	E-cycle (not) E-ch select unit 2	15. 49. 04
Use B-ch nu	Output cycle Output field cycle	15. 49. 05
E-ch select odd parity unit	E-ch select unit 4	13. 60. 03
Odd parity cycle	E-cycle E-ch select odd parity unit	13. 60. 02
Use B-ch zones	Output cycle Output field cycle Odd parity cycle	15. 49. 05
Gate asm ch to E1 input	E-ch output mode (not) Control reg disable	15. 62. 07

10. Set E1 data register.

<u>Signal</u>	<u>Control</u>	<u>Logic</u>
Set E1 reg	1401 card print-in-proc (not) E-ch 2nd addr trf	15. 62. 04
E1 reg full	Set E1 reg	15. 41. 10

11. Transfer E1 to E2.

<u>Signal</u>	<u>Control</u>	<u>Logic</u>
E-ch move mode	1401 card print-in-process	15. 62. 02
E-ch reset	Card print reset	15. 41. 12
E2 reg full latch (off)	E-ch reset	15. 41. 10
Set E2 reg	E1 reg full latch (on) E2 reg full latch (off)	15. 62. 04
E2 reg full latch (on)	E-ch output mode Set E2 reg	15. 41. 10
E1 reg full latch (off)	(not) Set E1 reg	15. 41. 10

When the E1 register full latch is turned off, E-cycle Required is brought up to cause another character to read out and be gated to the E1 register.

12. Request a punch-transfer scan.

<u>Signal</u>	<u>Control</u>	<u>Logic</u>
Ready to buffer	E-ch-in-process E-ch select any buffer E2 reg full	13. 70. 04
Ready	Ready to buffer	51. 40. 04
AC set pch trans req trig	Pch ready Ready (not) Pch busy	52. 10. 14
Pch trans req	Select unit 4 AC set pch trans req trig	52. 10. 15
Pch priority gate	Pch trans req	52. 10. 15
Pch pr req (trigger)	Pch priority gate Time 080-090	52. 10. 01
Gate on pch scan TR	Pch pr req Same scan (not) Inh rd priority req	52. 10. 01
Pch scan	Gate on pch scan TR Time 030-040	52. 10. 02
Pch trans scan	Pch ready Pch trans req Pch scan	52. 10. 06
Trans scan	Pch trans scan	51. 40. 40

13. Write the E2 character into the addressed buffer position.

<u>Signal</u>	<u>Control</u>	<u>Logic</u>
CPU to I/O Sync___bit	E2 reg___bit	15. 60. 31-34
CPU to I/O bit___	CPU to I/O sync___bit	51. 40. 10
Inh gate___bit	CPU to I/O bit___	51. 16. 03
Write pulse	070-100	51. 30. 05

14. Set the strobe trigger.

<u>Signal</u>	<u>Control</u>	<u>Logic</u>
Trans scan	Pch trans scan	51. 40. 40
Strobe latch	Trans scan Time 020-030	51. 40. 43
CPU to I/O trans	Pch trans scan	51. 40. 40
Buffer strobe	Strobe latch CPU to I/O trans Time 100-000	51. 40. 43

<u>Signal</u>	<u>Control</u>	<u>Logic</u>
Reset E2 full latch	E2 reg full E-ch strobe trigger E-ch output mode	15. 62. 05
E2 reg full latch (off)	Reset E2 full latch	15. 41. 10
E-ch strobe trigger (off)	Reset E2 full latch E-ch output mode	15. 62. 03

15. After 80 memory cycles, stop scanning the buffer and develop the external end-of-transfer.

<u>Signal</u>	<u>Control</u>	<u>Logic</u>
End of scan	Tens ring AC set Tens ring 7	51. 32. 01
End-of-transfer, 1-latch (on)	Trans scan	51. 40. 12
End-of-transfer, 2-latch (on)	End-of-transfer,1-latch (on) (not) Trans scan Time 090-100	51. 40. 12
End-of-trans	End-of-transfer, 2-latch (on)	51. 40. 12
Buffer end-of-trans	End-of-trans	51. 40. 12
E-ch ext end-of-transfer	Buffer end of trans E-ch select any buffer 1st clock pulse	13. 42. 11
E-ch int end-of-transfer	E-ch ext end-of-transfer E-ch output mode E1 reg full E2 reg full 2nd clock pulse	13. 63. 01
E-ch int end-of-trf delayed	E-ch int end-of-transfer Logic gate Z	13. 65. 05
E-ch status sample B	E-ch int end-of-trf delayed E-ch ext end-of-transfer (not) E-ch status sample Logic gate Z, (not) 2nd clock pulse	13. 65. 05
E-ch second sample B	E-ch status sample B (not) 2nd clock pulse	13. 65. 05
E-ch status sample B delay	E-ch second sample B (not) 2nd clock pulse	13. 65. 05

16. Energize the punch clutch.

<u>Signal</u>	<u>Control</u>	<u>Logic</u>
Correct trans to buffer	E-ch status sample B-delay 1401 mode E-ch select any buffer	13. 70. 04



<u>Signal</u>	<u>Control</u>	<u>Logic</u>
Go	Correct trans to buffer	51. 40. 04
Punch feed gate	Go Select unit 4	52. 10. 14
Punch feed	Proc pch Pch feed gate	52. 10. 15

17. Advance scan ring to the end position.

<u>Signal</u>	<u>Control</u>	<u>Logic</u>
1401 I/O ring advance	E-ch status sample B-delayed	13. 70. 01
1401 I/O end pos	1401 punch trigger 1401 I/O ring advance	13. 70. 02
1401 card print-in-proc (off)	1401 I/O ring advance 1401 I/O end	13. 70. 02

18. Turn off the unoverlap-in-process latch and bring up the last execute cycle.

<u>Signal</u>	<u>Control</u>	<u>Logic</u>
(not) E-ch unovlp-in-process	E-ch status sample B-delay	13. 60. 04
Last execute cycle I/O	1401 I/O end (not) 1401 branch latch	13. 65. 07
Last execute cycle	Last execute cycle I/O	12. 12. 51

### 8. 8. 03 Print a Line

#### Op-Code Function (2)

The print Op code causes the 132 (or 100) characters in CPU core storage to transfer to print storage. After a correct transfer, the printer is signalled to begin printing the information just transferred.

#### Operation (Figures 8.8-1 and 8.8-4)

No check is made in the 1401 mode for an I/O interlock condition. The status indicators are reset, and have no effect on the operation. The 1401 print Op code is recognized in the same manner as described for read and punch. The E-channel and unit 2 are selected. The transfer of data takes place in the unoverlapped mode. The address 00201 is developed so that core-storage locations 00201 through 00332 (or 00300) can read out in sequence to print storage. E-cycles are taken as required to keep the E1 and E2 registers full. The BAR is modified by plus one on each E-cycle. Each time E2 is filled, the character is stored in the next sequential print storage position. When the print storage is filled, an E-channel External-End-of-Transfer signal develops. This results in a Go signal which starts the printing operation. It also results in advancing the scan ring to read position to test for an Op-register 1 condition.

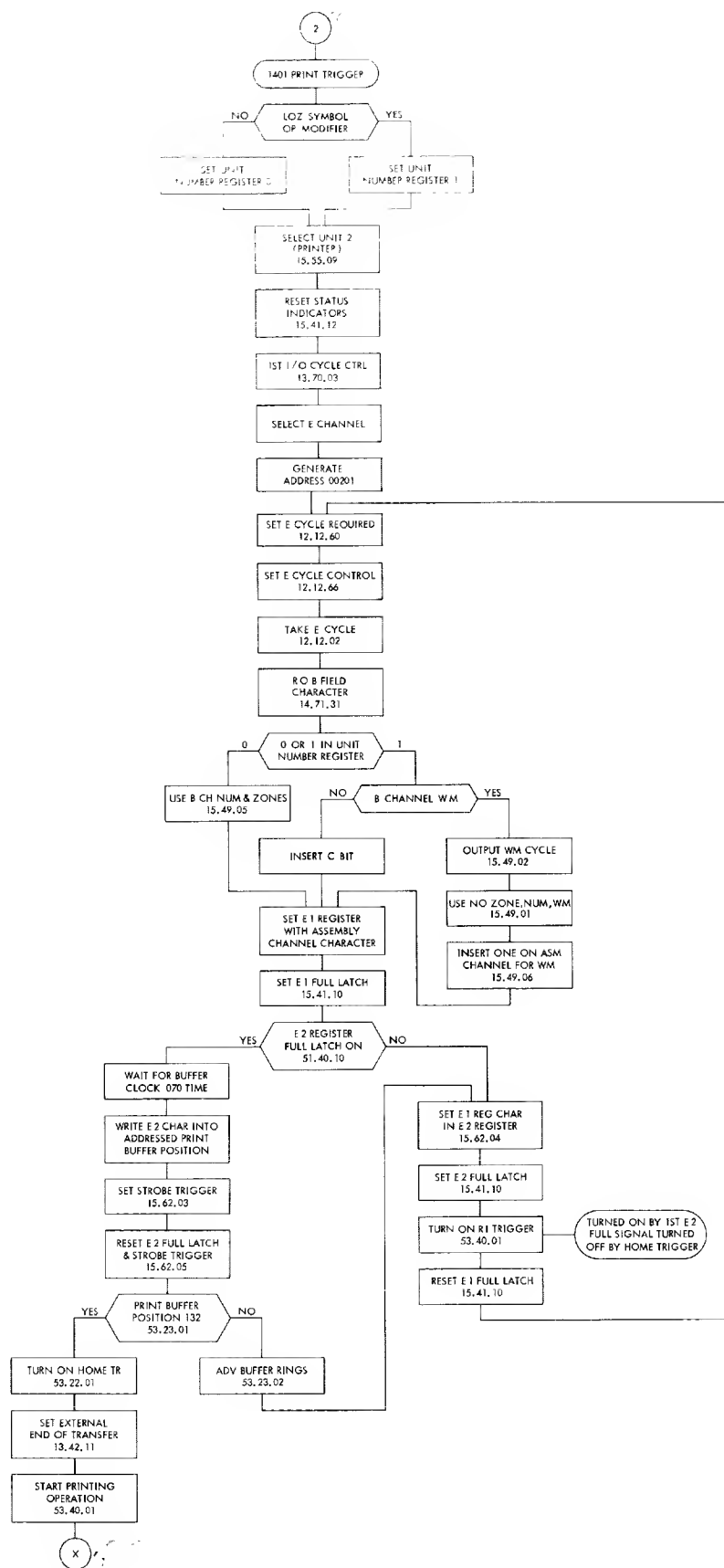


Figure 8.8-4 IBM 1401 Mode Print Op

## Circuits

### 1. Recognize a card-print operation.

<u>Signal</u>	<u>Control</u>	<u>Logic</u>
1401 card print ops	(not) Ctrl reg disable 1401 mode Op dcd: not A, not B, not 8	13. 13. 11

### 2. Scan for Print, Read, or Punch.

<u>Signal</u>	<u>Control</u>	<u>Logic</u>
1401 print trigger	1401 card print ops Last insn RO cycle Logic gate F	13. 70. 01
1401 print	1401 print trigger Op reg 2-bit	13. 70. 03

The 1401 print-signal blocks the drive signal for the scan ring. Thus, the 1401 print trigger remains ON until print storage is filled from core storage.

### 3. Signal Card Print-in-Process.

<u>Signal</u>	<u>Control</u>	<u>Logic</u>
1401 set cd print-in-proc	1401 card or print op code Last insn RO cycle Logic gate E	13. 70. 01
1401 card print-in-proc	1401 set cd print-in-proc	13. 70. 02

### 4. Select E-channel and Unit 2.

<u>Signal</u>	<u>Control</u>	<u>Logic</u>
E-ch select unit 2	1401 print	13. 50. 03
Unit 2, select to I/O	E-ch select unit 2	13. 50. 03
E-ch select any buffer	E-ch select unit 2	13. 60. 03
E-ch out mode (trigger)	1401 print trigger	15. 62. 01
E-ch output op to buffer	E-ch out mode	15. 62. 01
E-ch output mode	E-ch out mode (not) Control reg disable	15. 62. 01

### 5. Set up the E-channel for processing.

<u>Signal</u>	<u>Control</u>	<u>Logic</u>
Card print reset	1401 print 2nd clock pulse clamped	13. 70. 03
1st I/O cycle control	Card print reset 2nd clock pulse clamped	13. 70. 03

<u>Signal</u>	<u>Control</u>	<u>Logic</u>
E-ch unovlp-in-process	1401 card print-in-process 1st I/O cycle control	13. 60. 04
E-ch in-process	E-ch unovlp-in-process	13. 60. 04

6. Generate address 00201.

<u>Signal</u>	<u>Control</u>	<u>Logic</u>
RO 00201 index addr	1st I/O cycle control 1401 print trigger	14. 70. 14
RO fixed addr	RO 00201 index addr	14. 70. 14
AR bus (00201)	RO fixed addr and RO 00201 index addr	14. 15. 01 to 14. 15. 12

7. Request an E-cycle to fill E1.

<u>Signal</u>	<u>Control</u>	<u>Logic</u>
E-cycle required	E-ch output mode (not) E-ch int end-of-transfer E1 reg full latch (off) (not) E-cycle any last gate	12. 12. 62
(start logic clock)	E-cycle required E-ch unovlp-in-process	11. 10. 10
E-cycle ctrl	E-cycle required 2nd-clock pulse Logic gate Z (not) E-ch int end-of-transfer (not) F-cycle required	12. 12. 66
E-cycle	E-cycle ctrl Logic gate B or S	12. 12. 66

8. Read out the B-address to the B-channel.

<u>Signal</u>	<u>Control</u>	<u>Logic</u>
B or E or F-cycle ctrl	E-cycle ctrl	12. 12. 02
B or E or F-cycle	B-cycle	12. 12. 07
RO BAR	E-cycle ctrl E-ch unovlp-in-process	14. 71. 31
Reset BAR	B or E or F-cycle ctrl LG early B	14. 71. 21
Set BAR	B or E or F-cycle ctrl Logic gate B or C	14. 71. 11
Addr mod set to plus-one	E-cycle ctrl	14. 71. 41

9. Gate B-channel character to E1 input.

<u>Signal</u>	<u>Control</u>	<u>Logic</u>
Output cycle	E-ch output mode E-cycle	13. 60. 06
E-ch unit number 0	1401 print trigger (not) Loz symbol op modifier	15. 55. 04
Output field cycle	E-cycle E-ch select unit 2 E-ch unit number 0	15. 49. 04
Use B-ch nu	Output cycle Output field cycle	15. 49. 05
E-ch select odd parity unit	E-ch select unit 2	13. 60. 03
Odd parity cycle	E-cycle E-ch select odd parity unit	13. 60. 02
Use B-ch zones	Output-cycle Output field cycle Odd parity cycle	15. 49. 05
Gate asm ch to E1 input	E-ch output mode (not) Control reg disable	15. 62. 07

10. Set the E1 data register.

<u>Signal</u>	<u>Control</u>	<u>Logic</u>
Set E1 reg	1401 card print-in-proc (not) E-ch 2nd-addr-trf	15. 62. 04
E1 reg full	Set E1 reg	15. 41. 10

11. Transfer E1 to E2.

<u>Signal</u>	<u>Control</u>	<u>Logic</u>
E-ch move mode	1401 card print-in-process	15. 62. 02
E-ch reset	Card print reset	15. 41. 12
E2 reg full latch (off)	E-ch reset	15. 41. 10
Set E2 reg	E1 reg-full latch(on) E2 reg-full latch(off)	15. 62. 04
E2 reg full latch (on)	E-ch output mode Sct E2 reg	15. 41. 10
E1 reg full latch (off)	(not) Sct E1 reg	15. 41. 10

When the E1 reg full latch is turned off, E-cycle required is brought up to cause another character to be read out and gated to the E1 register.

12. Request a print transfer scan.

<u>Signal</u>	<u>Control</u>	<u>Logic</u>
Ready to buffer	E-ch in-process E-ch select any buffer E2 reg full	13. 70. 04
Ready	Ready to buffer	51. 40. 04
Read in	Ready Ready for CPU transfer	53. 40. 01
Print transfer	Read in (not) CE mode (not) C-bit insert	53. 11. 05
Trans scan	Print transfer	51. 40. 40

13. Write the E2 character into the addressed print-buffer position.

<u>Signal</u>	<u>Control</u>	<u>Logic</u>
CPU to I/O sync___bit	E2 reg___bit	15. 60. 31-34
CPU to I/O bit___	CPU to I/O sync___bit	51. 40. 10
CPU transfer	Print trans	53. 11. 05
Gated CPU input___bit	CPU transfer CPU to I/O bit___	53. 11. 01
Print data input___bit	Gated CPU input___bit	53. 11. 03
Inhibit gate___bit	Print data input___bit	53. 10. 05
Inhibit pulse 1, 2	Inhibit pulse Read in or scan (not) delay	53. 45. 01
___bit inh dr	Inhibit gate___bit Inhibit pulse 1, 2	53. 10. 06
Write pulse	Write pulse from I/O Read in or scan (not) delay	53. 45. 01

14. Sent the strobe pulse to the CPU to reset the E2 full latch.

<u>Signal</u>	<u>Control</u>	<u>Logic</u>
Trans scan	Print trans	51. 40. 40
Strobe latch	Trans scan Time 020-030	51. 40. 43

<u>Signal</u>	<u>Control</u>	<u>Logic</u>
CPU to I/O trans	Print trans	51. 40. 40
Buffer strobe	Strobe latch CPU to I/O trans Time 100-000	51. 40. 43
Reset E2 full latch	E2 reg full E-ch strobe trigger E-ch output mode	15. 62. 05
E2 reg full latch (off)	Reset E2 full latch	15. 41. 10
E-ch strobe trigger (off)	Reset E2 full latch E-ch output mode	15. 62. 03

15. After 132 memory cycles, turn on the home trigger.

<u>Signal</u>	<u>Control</u>	<u>Logic</u>
Turn off rings latch (on)	Read in	53. 23. 01
Gate on home trigger	Turn off rings latch (on) Threes-ring 3 Fives-ring 5 Tens-ring 3	53. 23. 01
Threes-ring advance	Read in Ring advance (time 010-020)	53. 23. 02
Read in (off)	Gate on home trigger Ring advance (time 010-020)	53. 40. 01
Home trigger	Threes-ring advance Gate on home trigger	53. 22. 01

16. Signal the external end-of-transfer, and develop the E-ch status sample B-pulse.

<u>Signal</u>	<u>Control</u>	<u>Logic</u>
End-of-transfer 1 latch	Trans scan	51. 40. 12
(not) Print trans	Read in (off)	53. 11. 05
(not) Trans scan	(not) Print trans	51. 40. 40
End-of-transfer 2 latch	End-of-transfer 1 (not) Trans scan Time 090-100	51. 40. 12
Buffer end-of-transfer	End-of-transfer 2	51. 40. 12
E-ch ext end-of-transfer	Buffer end-of-transfer E-ch select any buffer 1st clock pulse	13. 42. 11

<u>Signal</u>	<u>Control</u>	<u>Logic</u>
E-ch int end-of-transfer	E-ch ext end-of-transfer E-ch output mode E1 reg full E2 reg full	13. 63. 01
E-ch int end-of-trf delayed	E-ch int end-of-transfer Logic gate Z	13. 65. 05
E-ch status sample B	E-ch int end-of-trf delayed (not) 2nd clock pulse E-ch ext end-of-transfer Logic gate Z	13. 65. 05
E-ch second sample B	E-ch status sample B	13. 65. 05
E-ch status sample B-delay	(not) 2nd clock pulse E-ch second sample B (not) 2nd clock pulse	13. 65. 05

17. Start the printing operation.

<u>Signal</u>	<u>Control</u>	<u>Logic</u>
Correct transfer to buffer	E-ch status sample B-delay 1401 mode E-ch select any buffer	13. 70. 04
Go	Correct transfer to buffer	51. 40. 04
Scan call	Go Ready for print or forms Select unit 2 (not) Scan call	53. 40. 01

18. Set last execute cycle. The E-channel status-sample B-delayed signal advances the scan ring through the read and punch positions to turn ON 1401 I/O end.

<u>Signal</u>	<u>Control</u>	<u>Logic</u>
E-ch unovlp-in-process (off)	E-ch status sample B-delay	13. 60. 04
Last execute cycle I/O	1401 I/O end (not) 1401 branch latch	13. 65. 07
Last execute cycle	Last execute cycle I/O	12. 12. 51

#### 8.8.04 Write Word Marks

##### Op-Code Function (2'□)

The word marks in the print area of core storage print as 1's in the corresponding print positions. The lozenge symbol Op modifier translates the word marks to 1's and suppresses the printing of all zone, WM or numerical characters (except 1's for word marks).



Operation (Figures 8.8-1 and 8.8-4)

This Op code functions in the same manner as the write-a-line Op code. B-channel characters are gated through assembly and are analyzed for WM's. A 1-bit is inserted on the assembly channel each time a WM is sensed. All characters without word marks are converted to valid blanks (C-bit only).

#### Circuits

1. Develop an output WM cycle.

<u>Signal</u>	<u>Control</u>	<u>Logic</u>
Loz symbol op modifier	Op mod reg C-bit 4,not 2,not 1, op mod B A 8 op mod	13. 21. 21
E-ch unit number 1	Loz symbol op mod 1401 print trigger	15. 55. 04
Output WM cycle	E-ch select unit 2 E-cycle E-ch unit number 1	15. 49. 02

2. Insert one bit for each character containing a word mark.

<u>Signal</u>	<u>Control</u>	<u>Logic</u>
Assembly ch nu one insert	Output WM cycle B-ch WM bit Output cycle	15. 49. 06

3. Insert a C-bit for each character containing no word mark.

<u>Signal</u>	<u>Control</u>	<u>Logic</u>
Use no numerics	Output WM cycle B-ch not WM bit Output cycle	15. 49. 01
Use no zones	Output WM cycle Output cycle	15. 49. 01
Use no WM	Output cycle Move cycle	15. 49. 01

Use No Numerics and Use No Zones combine with B-Ch Not WM Bit to produce a C-bit.

### 8.8.05 Combination Card-Print Instructions

#### Op-Code Function

Two or three I/O Op code functions can be combined into one instruction. In any case, the printer takes priority over the reader, and the reader over the punch. The combination Op codes are as follows:

<u>Combination Op Codes</u>	<u>Op Code</u>	<u>Op Reg Bits</u>
Write and read	3	12
Read and punch	5	14
Write and punch	6	24
Write, read, and punch	7	124

#### Operation (Figures 8.8-1, 8.8-2, 8.8-3, and 8.8-4)

In each of the cases, the transfer between CPU core storage and the I/O storage unit takes place in the priority order print, read, punch. As soon as one transfer is complete, the corresponding unit is set in motion, and a test is made of the Op register bits to determine whether the next operation in priority order is to be performed. When a transfer-to-print-storage is complete, a test is made of the Op register 1-bit line (read). If it is up, a transfer of read storage to CPU core storage is made. If the Op register 1 line is down, an immediate test is made of the Op register 4-line to determine whether a punch operation should take place. If no Op register 4-bit exists, an I/O end signal results. This in turn causes a last execute cycle. The testing of the Op register bit output is made by the 4-position, 1401 I/O scan ring. This scans each Op register bit in turn to determine which operations should take place. The positions of the ring test the Op register outputs as follows:

<u>Position</u>	<u>Name of Trigger</u>	<u>Op Reg Bit Tested</u>	<u>Logic</u>
1	Print trigger	2	13.70.01
2	Read trigger	1	13.70.01
3	Punch trigger	4	13.70.02
4	1401 I/O end	-	13.70.02

The ring advances by a clock pulse, if the Op register bit line tested is not up. E-Ch Status Sample B Delayed advances the ring at the end of the data transfer, if the Op register bit line tested is on.

### 8.8.06 Select Stacker

#### Op-Code Function

This instruction selects the card that was just read or punched into the stacker pocket specified by the Op modifier.

<u>d-Character</u>	<u>Feed</u>	<u>Stacker</u>
1	Read	1
2	Read	8/2
4	Punch	4
8	Punch	8/2

#### Operation (Figure 8.8-5)

The select stacker Op code switches the Op-modifier character to the A-data register during the I-cycle in which it appears on the B-channel. The A-data register contents then transfer via E1 and E2 to the CPU to I/O bit lines. The forms stacker Go signal develops and is switched with stacker select to develop a signal that samples the CPU-to-I/O bit lines (Op modifier contents) to turn on either the read-stacker-select or the punch-stacker-select latches. Actual selection of the card in the punch feed does not occur until the punch busy goes OFF after the previous punching operation.

#### Circuits

Op-code grouping lines are:

1401 no exe cy branch ops  
 2-char only op codes  
 Op mod to A-ch on B-cycle ops  
 Regen mem on B-cycle ops  
 No branch

#### 1. Develop the stacker select signals.

<u>Signal</u>	<u>Control</u>	<u>Logic</u>
1401 stacker select op code	Op reg 1401 C-bit Op dcd B, not A, not 8 Op dcd 2, not 4, not 1	13. 13. 11
Stacker sel op code	1401 stacker sel op code	13. 13. 08
Stacker select to buffer	Stacker sel op code	13. 70. 04

#### 2. Reset the status indicators.

<u>Signal</u>	<u>Control</u>	<u>Logic</u>
For K E-ch reset	2-char-only op codes Logic gate C I-ring 1	15. 41. 12
E-ch reset This causes the reset of:	For K E-ch reset	15. 41. 12

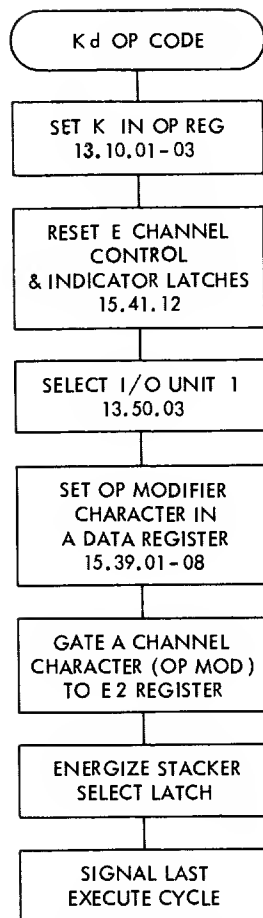


Figure 8.8-5 IBM 1401 Mode Stacker Select Op

<u>Signal</u>	<u>Control</u>	<u>Logic</u>
	E-ch data check	12. 62. 04
	E-ch not ready	12. 62. 01
	E-ch condition	12. 62. 04
	E-ch busy	12. 62. 02
	E-ch no transfer latch	13. 72. 04
	E-ch wrong-length record	13. 63. 03
	E-ch correct-length record	13. 63. 03

3. Set the Op modifier character in the A-data register.

<u>Signal</u>	<u>Control</u>	<u>Logic</u>
Sw B-ch to A-reg	I-cycle Logic gate D 2nd clock pulse	15. 38. 01

4. Gate the A-channel character (Op modifier) to the E2 register.

<u>Signal</u>	<u>Control</u>	<u>Logic</u>
A-ch ___ bit	Sw B-ch to A-reg	15. 39. 01-08
Use A-ch nu	I-cycle	15. 49. 03
Use A-ch zone	I-cycle	15. 49. 03
Asm ch ___ bit	Use A-ch nu	15. 40. 01-10
Gate asm ch to E1	I-cycle and not CR disable	15. 62. 07
Set E1 reg	2-char only op code A-ch not WM bit I-cycle	15. 62. 04
Copy E1 BCD to E2	(not) E1 reg word separator	15. 62. 06
Set E2 reg	E1 reg full (not) E2 reg full	15. 62. 04
E-ch ___ bit	Set E2 reg	15. 60. 31-34

5. Select the stacker magnet according to the E-channel (Op modifier) character.

<u>Signal</u>	<u>Control</u>	<u>Logic</u>
Forms stacker go	2-char only op codes Last insn RO cycle (not) E-ch not ready (not) E-ch busy (not) E-ch no transfer latch	13. 70. 04

<u>Signal</u>	<u>Control</u>	<u>Logic</u>
Stacker select	Stack select to buffer	51. 40. 01
Rd stk sel and	1401 mode	51. 40. 50
Pch stk sel	Stacker select Forms stacker go	
Rd stk one (or two) latch	Rd stk sel CPU to I/O bit 1 (or 2)	52. 13. 02
Stack four (or eight)	Pch stk sel CPU to I/O bit 4 (or 8)	52. 13. 01

The AC set for the stack 4 or 8 RD unit occurs when the select unit 4-line is up and punch busy goes OFF. This assures that the card is punched and ready for selection into the stacker pocket when the select magnet is energized.

#### 6. Signal the last execute cycle.

<u>Signal</u>	<u>Control</u>	<u>Logic</u>
Last execute cycle I/O	2-char only op codes 1401 mode (not) 1401 branch latch Last insn RO cycle E-ch any status(on)	13. 65. 07

### 8. 8. 07 Carriage Control

#### Op-Code and Op-Mod Functions

The F-Op code sends the Op-modifier character to the carriage circuitry to control forms skipping and spacing. The Op-modifier character causes the tape-controlled carriage to take a single, double, or triple space; or to skip to the next hole in the designated tape channel.

#### Operation (Figure 8. 8-6)

The forms control operation transfers the Op-modifier character to the A-data register during the I-cycle in which it appears on the B-channel. The A-data register contents then switch via E1 and E2 to the CPU I/O bit lines. The forms-stacker go signal then develops, and is switched with forms control to develop a CCC register gate signal that samples the CPU to I/O bit lines to set the CCC register.

#### Circuits

Op-code grouping lines are:

2-char only op codes  
1401 no exe cy branch ops  
Op-mod to A-ch on B-cy ops  
Regen mem on B-cy op codes

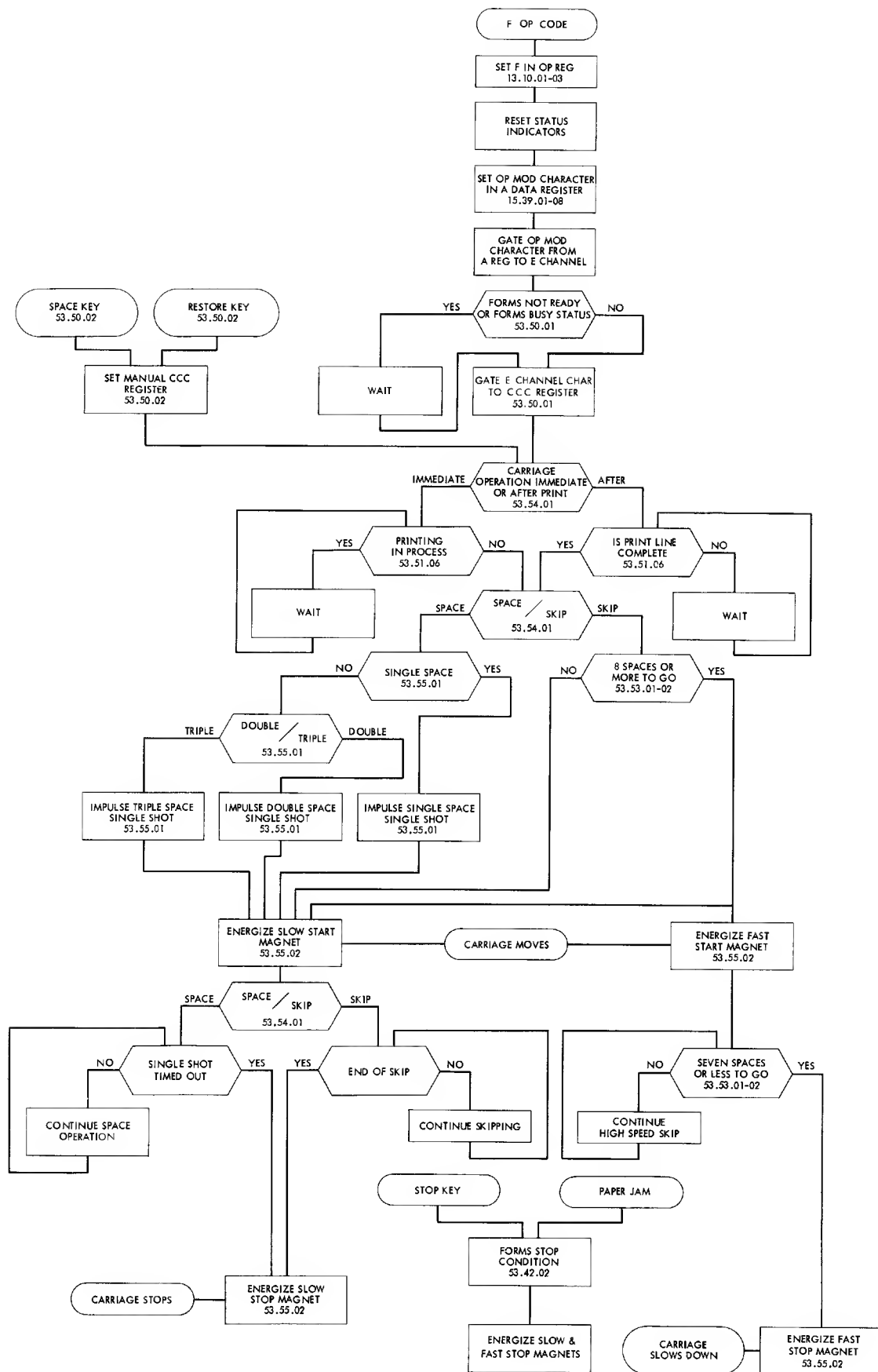


Figure 8.8-6 IBM 1401 Mode Carriage Control Op

1. Set up controls for forms Op.

<u>Signal</u>	<u>Control</u>	<u>Logic</u>
1401 forms ctrl	Op dcd B · A, not 8	13. 13. 11
Op code	Op dcd 42 not 1 Op reg 1401 C-bit	
Forms ctrl op code	1401 forms ctrl op code	13. 13. 08
Forms ctrl to buffer	Forms ctrl op code	13. 70. 04
Forms control	Forms ctrl to buffer	53. 50. 04

2. Reset the status indicators.

<u>Signal</u>	<u>Control</u>	<u>Logic</u>
For K E-ch reset	2-char only op codes Logic gate C I-ring 1	15. 41. 12
E-ch reset This causes the reset of:	For K E-ch reset  C-ch data check E-ch not ready C-ch condition E-ch busy E-ch no transfer latch C-ch wrong-length record E-ch correct-length record	15. 41. 12  12. 62. 04 12. 62. 01 12. 62. 04 12. 62. 02 13. 72. 04 13. 63. 03 13. 63. 03

3. Set the Op modifier character in the A-data register.

<u>Signal</u>	<u>Control</u>	<u>Logic</u>
Sw B-ch to A-reg	I-cycle Logic gate D 2nd clock pulse	15. 38. 01

4. Gate the A-channel character (Op modifier) to E2 register.

<u>Signal</u>	<u>Control</u>	<u>Logic</u>
A-ch__bit	Sw B-ch to A-reg	15. 39. 01-08
Use A-ch nu	I-cycle	15. 49. 03
Use A-ch zone	I-cycle (not) 1401 and I-ring 5 or 10 time	15. 49. 03
Asm ch__bit	Use A-ch nu Use A-ch zones	15. 50. 01-10
Gate asm ch to E1	I-cycle and not CR disable	15. 62. 07



<u>Signal</u>	<u>Control</u>	<u>Logic</u>
Set E1 reg	2-char only op code A-ch not WM bit I-cycle	15. 62. 04
Copy E1 BCD to E2	(not) E1 reg word separator	15. 62. 06
Set E2 reg	E1 reg full (not) E2 reg full	15. 62. 04
E-ch___bit	Set E2 reg	15. 60. 31-34

5. Gate the E2 register to the carriage-control character register.

<u>Signal</u>	<u>Control</u>	<u>Logic</u>
Forms stacker go	2-char only op codes Last insn RO cycle (not) E-ch not ready (not) E-ch busy (not) E-ch no transfer latch	13. 70. 04
Forms and stacker go	Forms stacker go	51. 40. 50
CCC reg gate	Ready for print or forms Forms control (not) Forms busy status	53. 50. 03
CPU to I/O___bit	E2 reg___bit	15. 60. 31-34
CCC reg___bit	CCC reg gate CPU to I___bit	53. 50. 01-02

6. Reset the CCC register. The CCC register resets in the usual manner, when the SS unit (53. 55. 03) times out at the end of carriage movement. The reset of the CCC register prevents repetition of the same operation.

The CCC register also resets at the beginning of each forms-control operation. The reset latch (53. 50. 03) controls the timing of this reset to occur between the time the forms-control line comes ON and the time the forms and stacker-go line comes ON. This reset allows the programmer to change the CCC register contents when a space-after-print or a skip-after-print has previously been set up. This change in the CCC register would occur while printing is taking place.

7. Develop magnet impulses for spacing and skipping. This occurs in the same manner as described in the section entitled 4. 4. 02 Forms Control Op Code.

8. Signal the last execute cycle.

<u>Signal</u>	<u>Control</u>	<u>Logic</u>
Last execute cycle I/O	2-char only op codes 1401 mode (not) 1401 branch latch Last insn RO cycle E-ch any status (on)	13. 65. 07

### 8. 8. 08 Card-Print-Branch Instructions

#### Op-Code Function

Any of the following 1401 input-output Op codes cause a branch to the I-address (read into the 1410 AAR) after the input or output data transfers.

<u>1401 I/O Op Codes</u>	<u>Op</u>	<u>I-Addr</u>	<u>Op Mod</u>
Read and branch	1	XXX	
Punch and branch	4	XXX	
Write and branch	2	XXX	
Write WM and branch	2	XXX	Loz
Read, punch, and branch	5	XXX	
Write, punch, and branch	6	XXX	
Write, read, punch, and branch	7	XXX	

#### Operation

Whenever the I-ring runs through 8-time, as it does with all of the codes in the preceding paragraph, the 1401 branch latch turns on. The operation then continues in the normal manner. The 1401 branch latch prevents the usual last-execute cycle, when 1401 I/O end comes ON. Instead, the 1401 branch latch switches with 1401 I/O end to cause a B-cycle. During this B-cycle the IAR (contains the address of the next instruction in sequence) reads out, is modified by zero, and returns to the BAR. The programmer can use the next following instruction to store the BAR, if it is needed later in the program. A last-execute cycle now occurs, and the next instruction is taken from the address in the AAR.

#### Circuits

1. Recognize the input-output branch Op.

<u>Signal</u>	<u>Control</u>	<u>Logic</u>
1401 I-ring-8 branch ops	1401 card or print ops	13. 13. 11

<u>Signal</u>	<u>Control</u>	<u>Logic</u>
1401 branch latch	1401 I-ring-8 branch ops I-ring-8 Last logic gate	12, 60, 19

2. Prevent normal last-execute cycle I/O. The 1401 branch latch (13. 65. 07) prevents the usual last-execute cycle I/O, when 1401 I/O end comes ON.
3. Transfer IAR to BAR after I/O transfers are complete.

<u>Signal</u>	<u>Control</u>	<u>Logic</u>
1401 I/O set branch cnds	1401 I/O end 1401 branch latch	13, 70, 02
B-cycle ctrl This causes a B-cycle in which <u>Set BAR</u> takes place.	1401 I/O set branch cnds	12, 12, 21
No scan ctrl This causes <u>Modify by Zero</u> .	1401 I/O set branch cnds	12, 30, 03
RO IAR The IAR reads out to be modified by zero, and then returns to the BAR.	No scan ctrl B-cycle ctrl LG spec A	14, 71, 34

4. Signal last execute cycle.

<u>Signal</u>	<u>Control</u>	<u>Logic</u>
Last-execute cycle br cnds	No scan B-cycle	12, 60, 08
Last-execute cycle	Last-execute cycle br cnds	12, 12, 51

5. RO AAR on 1st I-cycle.

<u>Signal</u>	<u>Control</u>	<u>Logic</u>
RO AAR	1401 branch latch 1401 mode I-cycle ctrl I-ring ctrl	14, 71, 30

#### 8. 8. 09 Stacker Select or Control Carriage and Branch

##### Op-Code Function

The stacker select and branch or control carriage and branch instructions operate

in the same manner as control carriage or select stacker except that the next instruction is taken from the specified I-address. The instruction format is either K (XXX) d, or F (XXX) d.

## Operation

If the I-ring goes to I9 (as it does when a modifier is included), the 1401 branch latch sets. At the same time, a 1401 take-I-to-B-cycle is forced. This causes an immediate B-cycle with no scan, during which the last instruction address that was used transfers from IAR to BAR for possible future use by the programmer. A last-execute cycle is forced during the B-cycle. During the resulting I-cycle, the 1401 branch latch causes the AAR contents to be placed in STAR. Both stack select and carriage control operations occur as described in previous sections.

## Circuits

Refer to previous sections for the basic circuits for either Op code. The following additional circuits are developed to cause the branch to the I-address (read into the 1410 AAR).

1. Recognize the stacker select or control carriage and branch operation.

<u>Signal</u>	<u>Control</u>	<u>Logic</u>
1401 I-ring-9 branch ops	1401 stacker select op code or 1401 forms ctrl	13. 13. 11
1401 branch latch	1401 I-ring-9 branch ops I-ring-9 time Last logic gate	12. 60. 19

2. Transfer IAR to BAR.

<u>Signal</u>	<u>Control</u>	<u>Logic</u>
1401 take I to B-cycle	1401 no exe cy branch ops 1401 I-ring-9 branch ops I-ring-9 Last insn RO cycle	12. 60. 19
Set B-cycle ctrl br ops	1401 take I to B-cycle	12. 60. 04
Set B-cycle ctrl	Set B-cycle ctrl br ops	12. 12. 44
B-cycle ctrl latch	Set B-cycle ctrl Last logic gate	12. 12. 21
Set no scan ctrl	1401 take I to B-cycle	12. 30. 05
RO IAR	No scan ctrl B-cycle ctrl Logic gate special A	14. 71. 34

The IAR reads out to be modified by zero and then returns to the BAR during the B-cycle so that it can later be stored by the programmer, if required.

### 3. Signal last-execute cycle.

<u>Signal</u>	<u>Control</u>	<u>Logic</u>
Last-execute cycle br ends	No scan B-cycle	12. 60. 08
Last-execute cycle	Last-execute cycle br ends	12. 12. 51

### 4. RO AAR on 1st I-cycle.

<u>Signal</u>	<u>Control</u>	<u>Logic</u>
RO AAR	1401 branch latch 1401 mode 1-cycle ctrl 1-ring ctrl	14. 71. 30

## 8. 8. 10 Start Read Feed and Start Punch Feed

### Op-Code Function (8, 9)

Both of these Op-codes are used on the 1401 to avoid interlock time. This provides more program time. Because the 1410 last-execute cycle occurs as soon as the I/O storage unit fills, no interlock time is involved. Thus, neither of these codes has any function on the 1410, and they are handled the same as a No Op.

### Operation

Logic 13. 13. 11 indicates the manner that codes 8 and 9 develop to cause a No Op signal. See the section entitled, 8. 7. 04 No Operation (N), for a description of the No Op function.

## 8. 8. 11 1401 I/O Error Conditions

### 1401 Read Error

A buffer error signal turns on the 1401 read error latch. The latch output can then be tested by the 1401 condition test Op code to cause a branch to a halt instruction, if desired.

### Circuits

<u>Signal</u>	<u>Control</u>	<u>Logic</u>
E-ch check bus	Buffer error E-ch sel any buffer	12. 62. 02

<u>Signal</u>	<u>Control</u>	<u>Logic</u>
1401 Read error	E-ch check bus 1401 Read trigger E-ch status sample B	13. 65. 01

#### 1401 Punch Error

Either a buffer-error or buffer-condition signal turns on the 1401 punch error latch at the end of the transfer to the buffer. The latch output can be tested by the programmer by a 1401 condition test operation to halt a branch instruction, if desired.

#### Circuits

<u>Signal</u>	<u>Control</u>	<u>Logic</u>
E-ch check bus	Buffer error E-ch sel any buffer	12. 62. 03
E-ch condition bus	Buffer condition E-ch sel any buffer	12. 62. 03
E-ch check or condition	E-ch check bus or E-ch condition bus	12. 62. 03
1401 Punch error	E-ch check or condition E-ch status sample B 1401 Punch trigger	13. 65. 01

#### 1401 Print Error

The 1401 print error latch is turned on at the end of the buffer transfer by either a buffer-error or buffer-condition signal. The programmer can then use a 1401 condition-test operation to branch to a halt instruction.

#### Circuits

<u>Signal</u>	<u>Control</u>	<u>Logic</u>
E-ch check or condition	E-ch check bus or E-ch condition bus	12. 62. 03
1401 Print error	E-ch check or condition E-ch status sample B 1401 print trigger	13. 65. 01

#### 1401 Inquiry Error

Any of the following errors during the console read or write operations turn on the 1401 inquiry error latch (13. 65. 08).

1. Invalid A-ch on a console read Op.
2. Console data check on a console write Op.
3. E2 register full (last character) when E-ch status sample B comes on.

The programmer can use this 1401 condition test Op to branch to an error routine or to halt instruction.

I/O Check Stop 12 / 5.04

The CE switch I/O check stop can be turned on to stop the machine at the last-execute cycle time, after a 1401 read, punch or print error.

#### Circuits

<u>Signal</u>	<u>Control</u>	<u>Logic</u>
1401 I/O error	1401 Read error or 1401 Punch error or 1401 Print error	13. 65. 01
Stop latch	1401 I/O error I/O chk st(on) 1401 mode Last-execute cycle	12. 12. 04

#### 1401 I/O Check Reset Switch

This switch prevents errors from setting the 1401 read, punch or print error latches (13. 65. 01).

A CH		B		A		ASSEM		OP		MOD ADDR	
CYC	SEL										
A			ER	ER		ER					ER
B											
C	A	WM	WM	WM	WM						
D	d	C	C	C	C			C	C		
E	E	B	B	B	B			B	B		
F	F	A	A	A	A			A	A		8
		8	8	8	8			8	8		4
		4	4	4	4			4	4		2
I		2	2	2	2			2	2		1
X		1	1	1	1			1	1		0

MATRIX	
H	X 1A
32	33

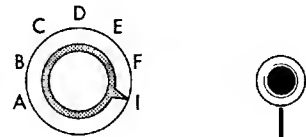
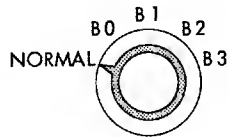
INDEX	
B	B
A	A

STOR ADDR REG				
8	8	8	8	8
4	4	4	4	4
2	2	2	2	2
1	1	1	1	1
0	0	0	0	0

ADDR STOP



B CH CHAR SEL



TRANSFER ADDRESS TO  
STORAGE ADDRESS REGISTER

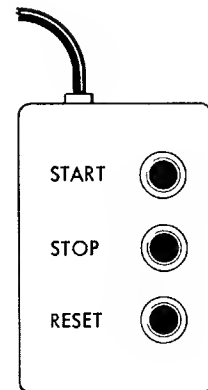
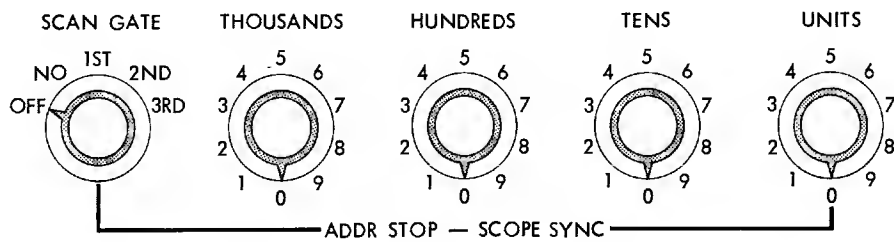


Figure 9.1-1 IBM 1411 CE Panel



## 9.0.00 SYSTEM CE CONTROLS

### 9.1.00 IBM 1411 CE CONSOLE (Figure 9.1-1)

#### 9.1.01 Address Display

Manual controls on the CE console display the addresses in the registers for checking purposes. A rotary switch selects the particular register for display. After the switch setting selects the register, pressing the address-display switch on the CE console starts the move and display operation. The selected-address register is gated out to the address bus and set into the STAR, where the latch outputs turn on the indicator.

Another control impulse, which the address-display switch initiates, inhibits the other registers that are not selected by the rotary address-selector switch.

#### 9.1.02 B-Character-Select Switch

The B-character-select switch permits gating out any one of the 4 positions of the B-data register to the B-channel indicator for visual checking. The switch must be reset to normal position (1) for normal operation or a B-character-select error occurs (15.30.01).

#### 9.1.03 Portable Controls

The portable controls plug into the six-position pin jack on the CE panel to provide a portable control for cycling the 1411 CPU.

This control permits single-cycle operation to be executed for CE checking purposes.

##### Start

The start key operation is normal.

##### Stop

The portable stop key parallels the console stop key in function.

##### Reset

The reset key is on the portable control unit. The control that this key initiates is parallel to the console computer-reset function.

#### 9.1.04 Address Stop and Scope Sync (Figure 9.1-2)

The storage-address stop switch and sync-control rotary switches provide manual control for selecting a particular address location to initiate a stop or a scope sync signal.

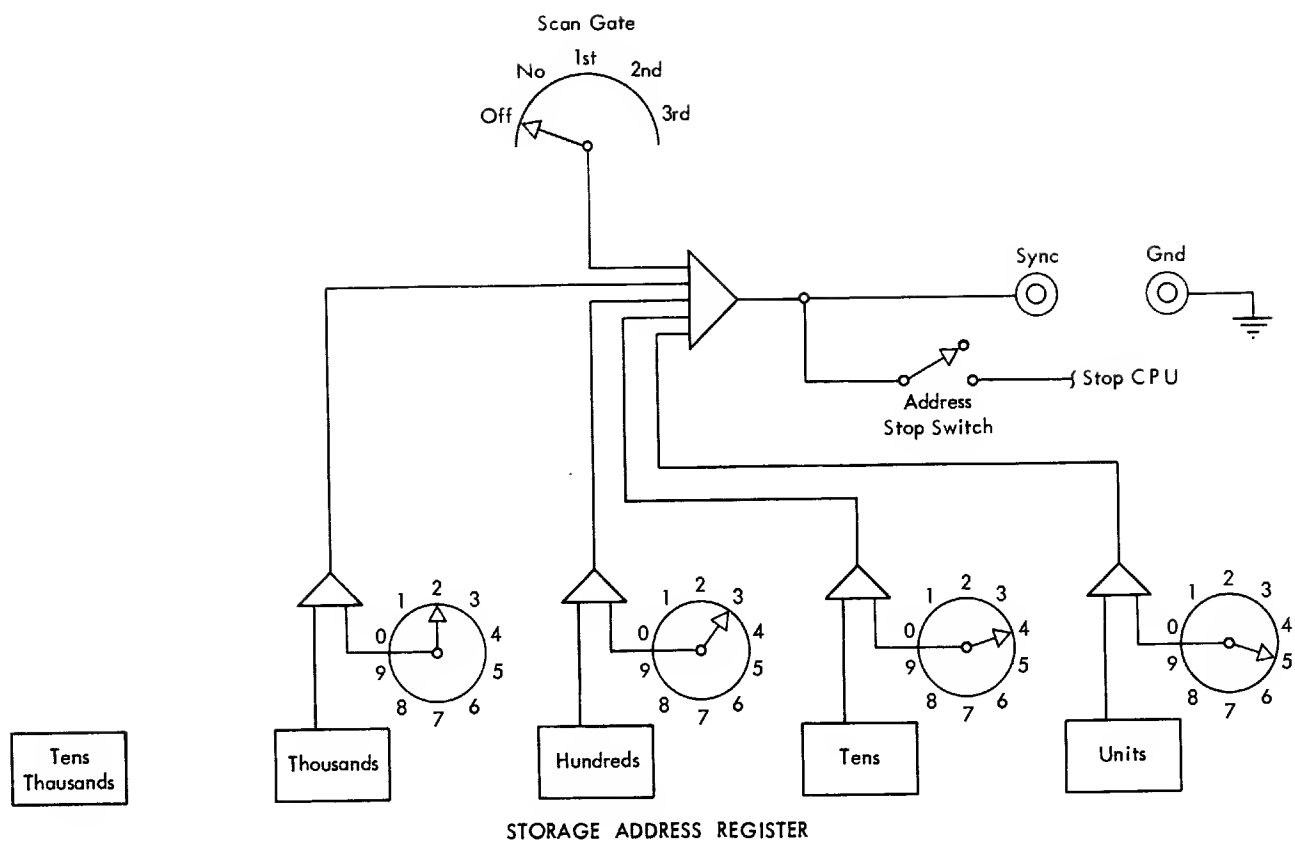


Figure 9.1-2 Address Stop or Scope Sync

A toggle switch controls the address stop for ON or OFF operation. The sync hub is effective at all times.

The address-selection switches are effective through the four low-order positions of the storage address register. The stop, or scope signal, can be further conditioned by a scan-gate switch for the scan mode desired (logic 14.17.17-19).

#### 9.2.00 IBM 1414 I/O SYNCHRONIZER CE PANEL

The I/O synchronizer CE panel gives the customer engineer full control of the integrated synchronizer and print storage. The panel allows him to remove various I/O units from CPU control so that he can test and service these units off line. Flexibility in the controls allows many diagnostic procedures to be set up as aids in trouble analysis.

##### 9.2.01 Switches, Keys, and Lights (Figure 9.2-1)

The diagnostic panel for the I/O synchronizer consists of two functional areas: an indicator panel and a switch panel. The first is made up of various lights that identify each character by bit structure, buffer address, and I/O machine. Error conditions are indicated. The second area contains the controlling switches and keys for the synchronizer.

##### Indicator Panel

**Integrated Buffer.** The lights in this group indicate which machine that is associated with the integrated buffer is in operation (logic 51.45.04).

**Buffer Addr.** These lights show the address of each character as it is scanned by the integrated buffer. In addition, a light indicates an end-of-scan condition (logic 51.45.01).

**Buffer Reg.** The bit structure of each character is displayed as it appears in the buffer register. Hole-count check-core status is indicated along with hole-count and parity errors (check search light). Lights also show ring-check and clock-check conditions (logic 51.45.02, 51.45.03).

**Print Address.** These lights give the print-buffer address of each character as it is scanned out to the printer. The buffer ring home position (home) is also shown (logic 51.45.05).

**Print Buffer.** This light shows that the print buffer is in operation (51.45.06).

**Print Reg.** The bit structure of each character, as well as conditions necessary for printing each character, is shown in logic 51.45.06.

##### Switch Panel

**Buffer Control.** Three keys provide start, stop and reset functions. The start key begins an operation when the normal synchronizer operation is modified by one of the diagnostic panel switches. Under normal operation, the 1411 CPU controls the entire system. Pressing the reset key resets some of

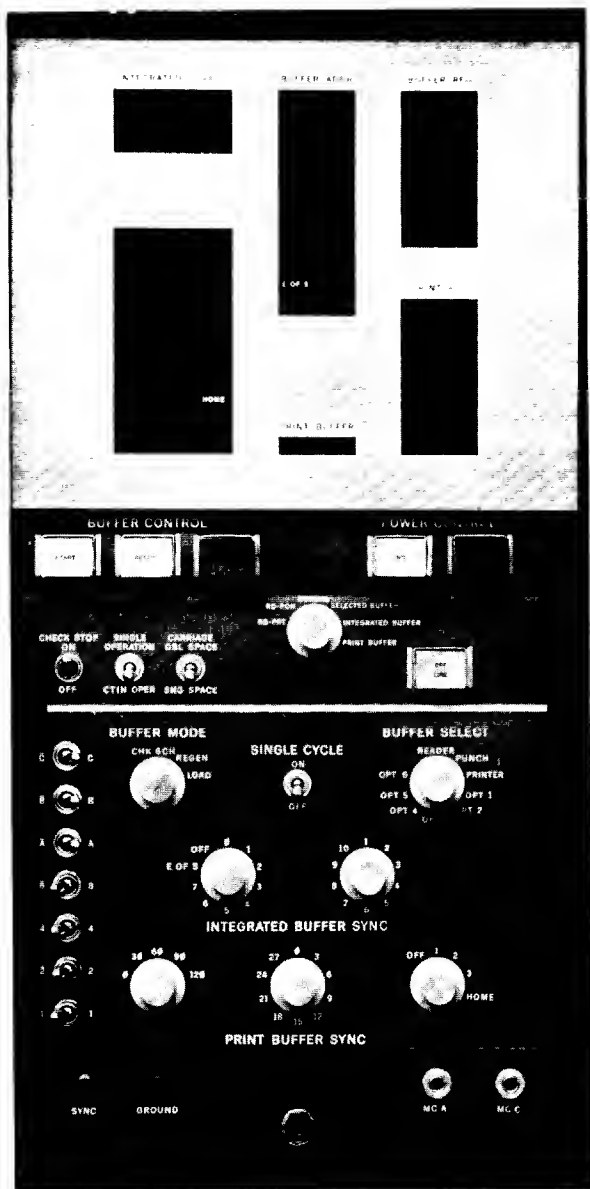


Figure 9.2-1 Input-Output Synchronizer CE Panel

the error detection circuits. Pressing the stop key stops the synchronizer after the operation in process ends.

**Power Control.** When the machine operates off line, two keys (labelled ON and OFF) control power to the synchronizer. When the machine operates on line, the IBM 1415 Console controls the power.

**Check Stop.** The check stop switch, when ON, stops the synchronizer after an operation during which an error was detected. With this switch OFF, the operation continues after an error.

**Op Control.** The operation control switch either operates the synchronizer in the mode indicated by the setting of the other switches, or performs a single operation each time the start key is pressed. The single operation can be a single scan, single card, single-print line, etc., as controlled by the control switch settings.

**Space.** This switch causes either single or double carriage spacing in the printer when it operates off line.

**Off-Line Mode.** The off-line mode switch selects the type of off-line operation to perform. This switch allows one or two units to be removed from the line without tying up the entire synchronizer. It is set to the normal position when all units are operating on line. Rd-Prt (reader-to-printer), or Rd-Pch (reader-to-punch) operations can be selected. The selected-buffer position allows selection of an individual buffer under control of the buffer-select switch. Only the selected buffer is removed from the line. The integrated-buffer and print-buffer positions remove the indicated units from the line for servicing.

**Off Line.** The off-line key removes the area selected by the off-line mode switch from CPU control. Pressing this key also activates the power-control on and off keys.

**Bit Entry.** These seven switches (C, B, A, 8, 4, 2, 1) are used to enter any character into any buffer position. They are active when in the selected-buffer, print-buffer, or integrated-buffer off-line modes.

**Buffer Mode.** This switch allows three types of operations when the machine is in the integrated-buffer or print-buffer off-line mode: load (to enter bit data from the bit switches), check search (to locate errors), and regen to regenerate the same data back into the buffer from which it came).

**Buffer Select.** The buffer-select switch, in conjunction with the selected-buffer setting of the off-line mode switch, selects a particular buffer to operate off line.

**Integrated Buffer Sync.** These switches develop a sync signal corresponding to the integrated buffer address. The sync signal is available at the sync hubs.

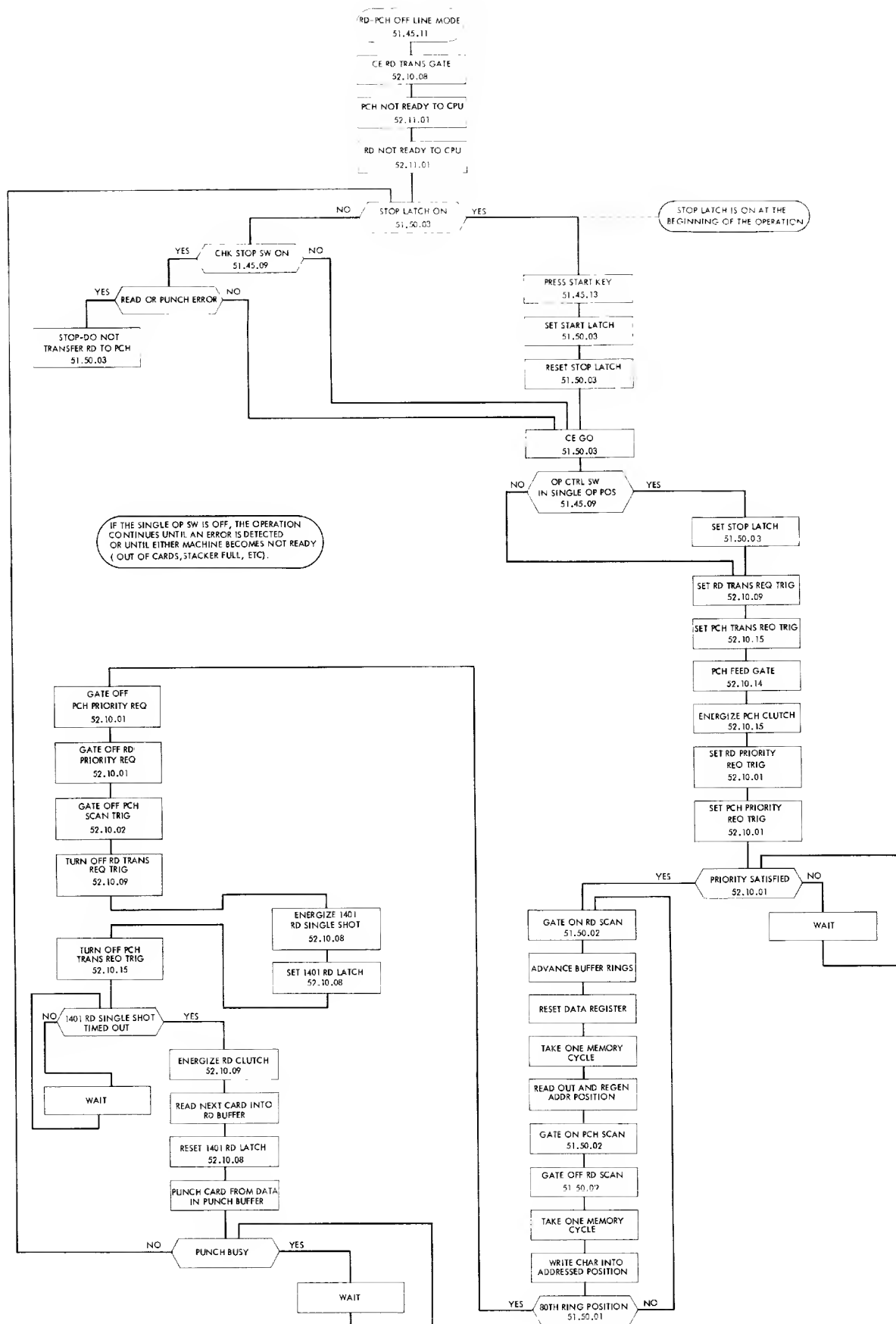


Figure 9.2-2 Card Reader to Punch Off-Line Mode

Print Buffer Sync. These switches develop a sync signal corresponding to the print-buffer address.

Sync. These hubs emit pulses to synchronize an oscilloscope during servicing. The sync switches define the pulses.

Remote Box Receptacle. When the remote box is plugged into the receptacle that is provided, it gives the customer engineer remote start, stop and reset functions.

MC-A, MC-C. These hubs provide marginal checking of frames 1414A and 1414C.

### 9.2.02 Read-to-Punch Off-Line Mode

#### Function

The read-punch off-line mode duplicates card for each card that feeds through the read feed.

The 1402 read-feed reads each card and places the information in the read buffer. From the read buffer, the record transfers to the punch buffer. After the transfer is complete, the 1402 punch feed punches a new card with the information just transferred to the punch buffer.

#### Operation (Figure 9.2-2)

To perform this operation, cards must be run into both read and punch feeds. This run-in causes the first card that feeds into the read feed to load into the read buffer and to establish the interlocks for both machines. After the run-in, the operation is set up as follows:

<u>Switch</u>	<u>Setting</u>	<u>Notes</u>
Off line	On	-
Off-line mode	Rd-pch	Removes the reader and the punch from CPU control.
Check stop	On	Stops the operation after the card in which an error occurs.
	Off	Allows errors to be ignored.
Op control	Single op	Stops the operation after each card feed.
	Ctin op	The operation continues until either machine becomes not-ready.
Carriage	-	No effect.
Buffer mode	-	No effect.
Single cycle	-	No effect.
Buffer select	-	No effect.
Bit switches	-	No effect.

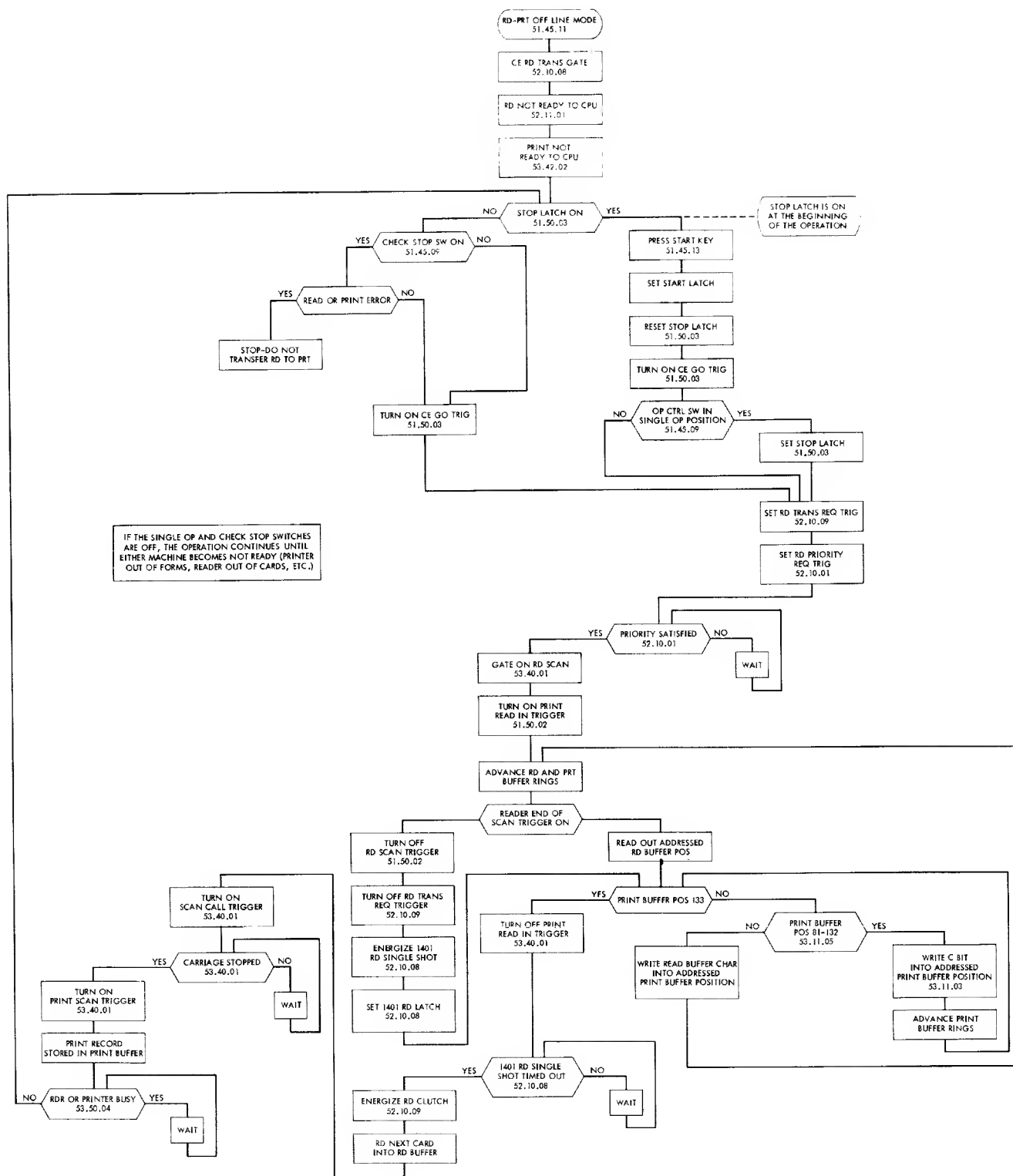


Figure 9.2-3 Card Reader to Printer Off-Line Mode



The buffer control start key must be pressed to start the operation. If the Op-control switch is in the single-Op position, one card is punched each time the start key is pressed. The transfer takes place via a series of single memory-cycle read and punch scans. A read-scan memory-cycle reads out the addressed read-buffer position, and sets this character into the data register. At the end of the read-scan memory-cycle, a punch-scan memory-cycle starts. The character in the data register writes into the addressed punch-buffer position. The data register resets, and the rings advance only at the beginning of each read-scan memory-cycle to allow two memory cycles (scans) to occur for each addressed position.

If an error stops the operation because the check-stop switch is ON, pressing the start key resets the check circuits before the next operation starts.

### 9.2.03 Read-to-Print Off-Line Mode

The read-to-print off-line mode causes the 1403 printer to print the information that the 1402 reader reads from cards. The operation is independent of the CPU. As the reader reads each card, the information from that card is placed in the read buffer. The contents of the read buffer then transfer to the print buffer and the line prints.

Operation (Figure 9.2-3)

Cards are run into the 1402 read feed to load the first card into the read buffer. After the run-in is complete, the operation is set up as follows:

<u>Switch</u>	<u>Setting</u>	<u>Notes</u>
Off line	On	-
Off-line mode	Rd-prt	Removes both the reader and the printer from CPU control.
Check stop	On	Stops the operation after the card in which an error occurs.
	Off	Ignores errors.
Op control	Single op	Stops the operation after each print line.
	Ctin op	The operation continues until either machine becomes not-ready.
Carriage	Sng space	Causes an automatic single space before each print line.
	Dbl space	Causes an automatic double space before each print line.
Buffer mode	-	No effect.
Single cycle	-	No effect.
Buffer select	-	No effect.
Bit switches	-	No effect.

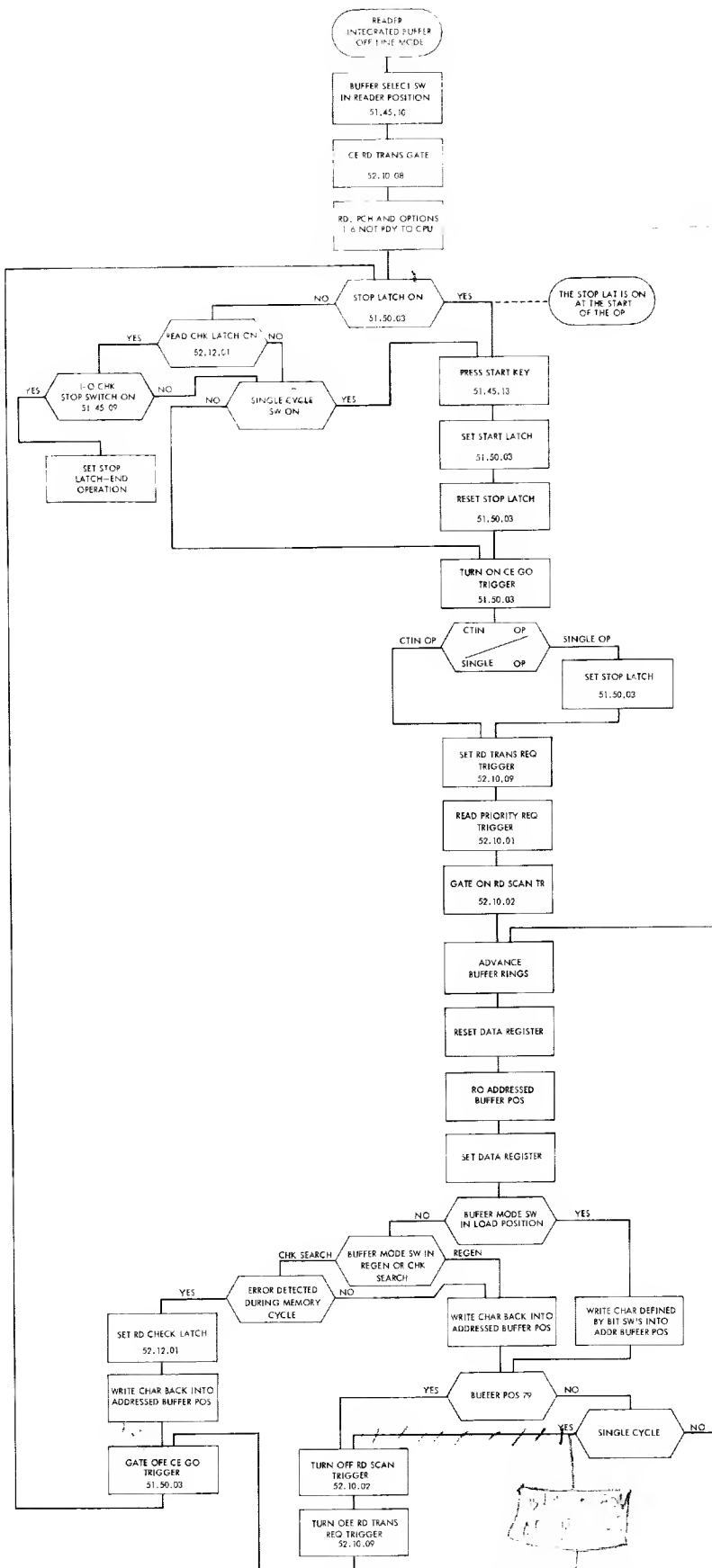


Figure 9.2-4 Card Read Integrated Buffer Off-Line Mode

Pressing the buffer-control start key starts the read-to-print operation. A read scan and a print scan start at the same time. Both read-buffer and print-buffer rings run simultaneously, because there is no memory-sharing between these two units. However, both buffers are driven by the same clock. During the read portion of a memory cycle, the addressed read-buffer position reads out and is gated to the inputs of the print-buffer inhibit-drivers. The character writes into the addressed print-buffer position during the write portion of the memory cycle. Both the integrated-buffer and print-buffer rings advance on each memory cycle.

After 80 memory cycles, the integrated-buffer end of scan-trigger signals the end of the read transfer. The print buffer then inserts C-bits in all positions from 81 through 132. When the print-buffer home-trigger signals that the print read-in is complete, the print operation starts. While the line is being printed, the reader reads in the next card. When both reader and printer finish their respective operations, another transfer takes place (under the control of the Op-control switch).

#### 9.2.04 Card-Read Integrated-Buffer Off-Line Mode

##### Function

The read integrated-buffer off-line mode allows the read buffer to be selected and scanned under control of the other switches on the CE panel. The card reader is not set in motion, and all units associated with the integrated buffer are removed from CPU control.

##### Operation (Figure 9.2-4)

The operation is set up as follows:

<u>Switch</u>	<u>Setting</u>	<u>Notes</u>
Off line	On	-
Off-line mode	Integrated buffer	-
Buffer select	Reader	-
Check stop	On	Stops the operation after the scan in which the error occurs.
	Off	Errors are ignored (see buffer-mode switch).
Op control	Single op	Stops the operation after each scan.
	Ctin op	Causes continuous scanning.
Carriage	-	No effect.
Buffer mode	Chk sch	The operation stops after the cycle in which the error occurs.
	Regen	Each position reads out and is regenerated.
	Load	Allows the characters that the bit switches define to enter into the buffer.

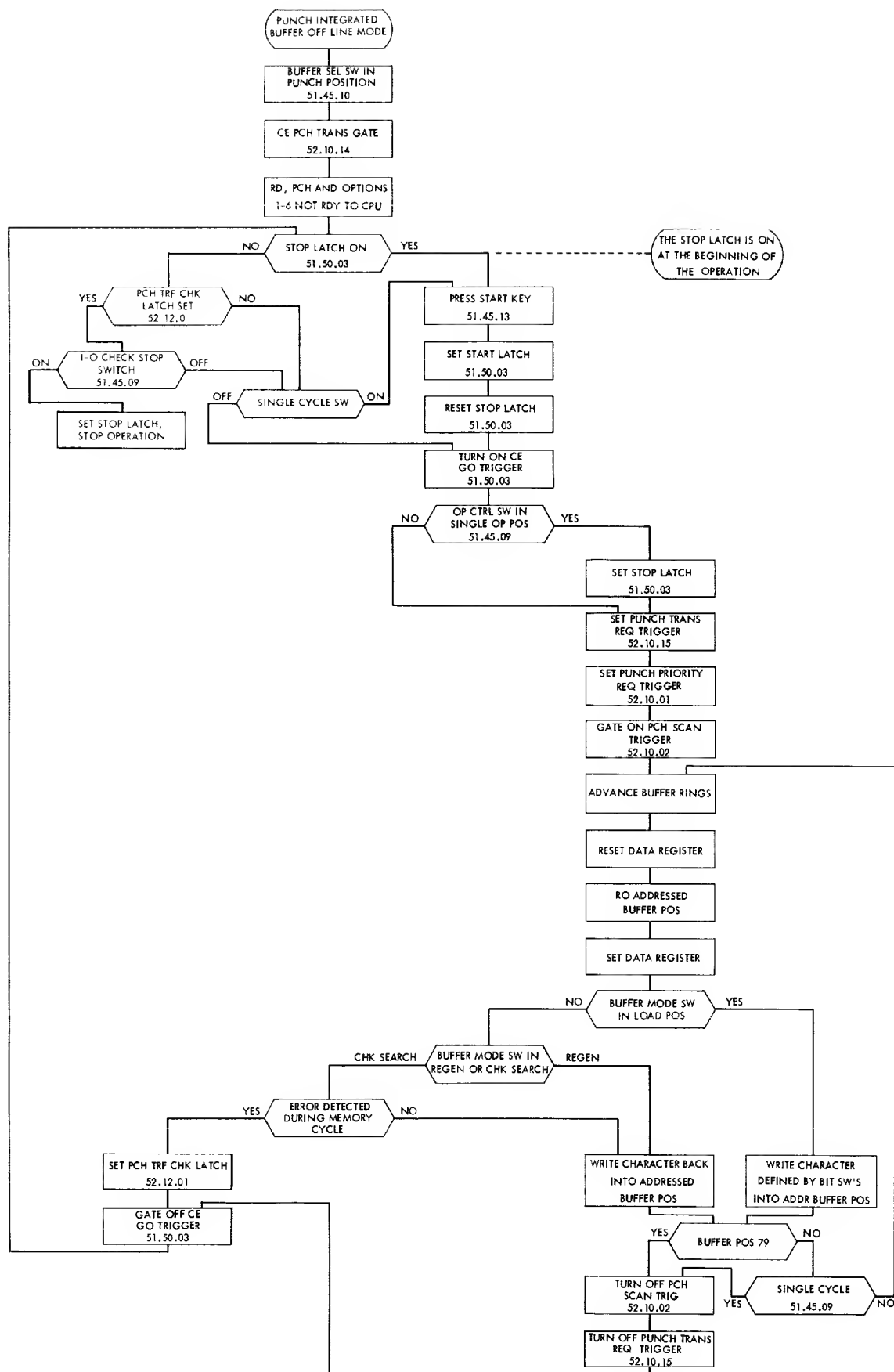


Figure 9.2-5 Punch Integrated Buffer Off-Line Mode

<u>Switch</u>	<u>Setting</u>	<u>Notes</u>
Single cycle	On	The operation stops after each memory cycle.
	Off	The operation is under the control of the op control switch.
Bit switches	Off or on	Inserts bits when the buffer-mode switch is in the load position

The start key is pressed to start any variation of this operation. Once the operation starts, the buffer mode, single cycle, and Op-control switches control when it stops.

#### 9.2.05 Card Punch Integrated-Buffer Off-Line Mode

##### Function

The punch integrated-buffer off-line mode selects and scans the punch buffer under control of the other switches on the CE panel. A scan can be started to search for errors. The buffer contents can be displayed. New information can be inserted into the punch buffer. The card punch is not set in motion. All units associated with the integrated buffer are removed from CPU control.

Operation (Figure 9.2-5)

The switch settings for this operation are:

<u>Switch</u>	<u>Setting</u>	<u>Notes</u>
Off line	On	-
Off-line mode	Integrated buffer	-
Buffer select	Punch	-
Check stop	On	Stops operation after the scan in which an error occurs.
	Off	Allows errors to be ignored (see buffer-mode switch).
Op control	Single op	Stops the operation after each scan.
	Ctin op	Causes continuous scanning.
Carriage	-	No effect.
Buffer mode	Chk sch	The operation stops after the cycle in which an error occurs.
	Regen	Each position reads out and is regenerated.
	Load	Allows the characters that the bit switches define to enter into the buffer.

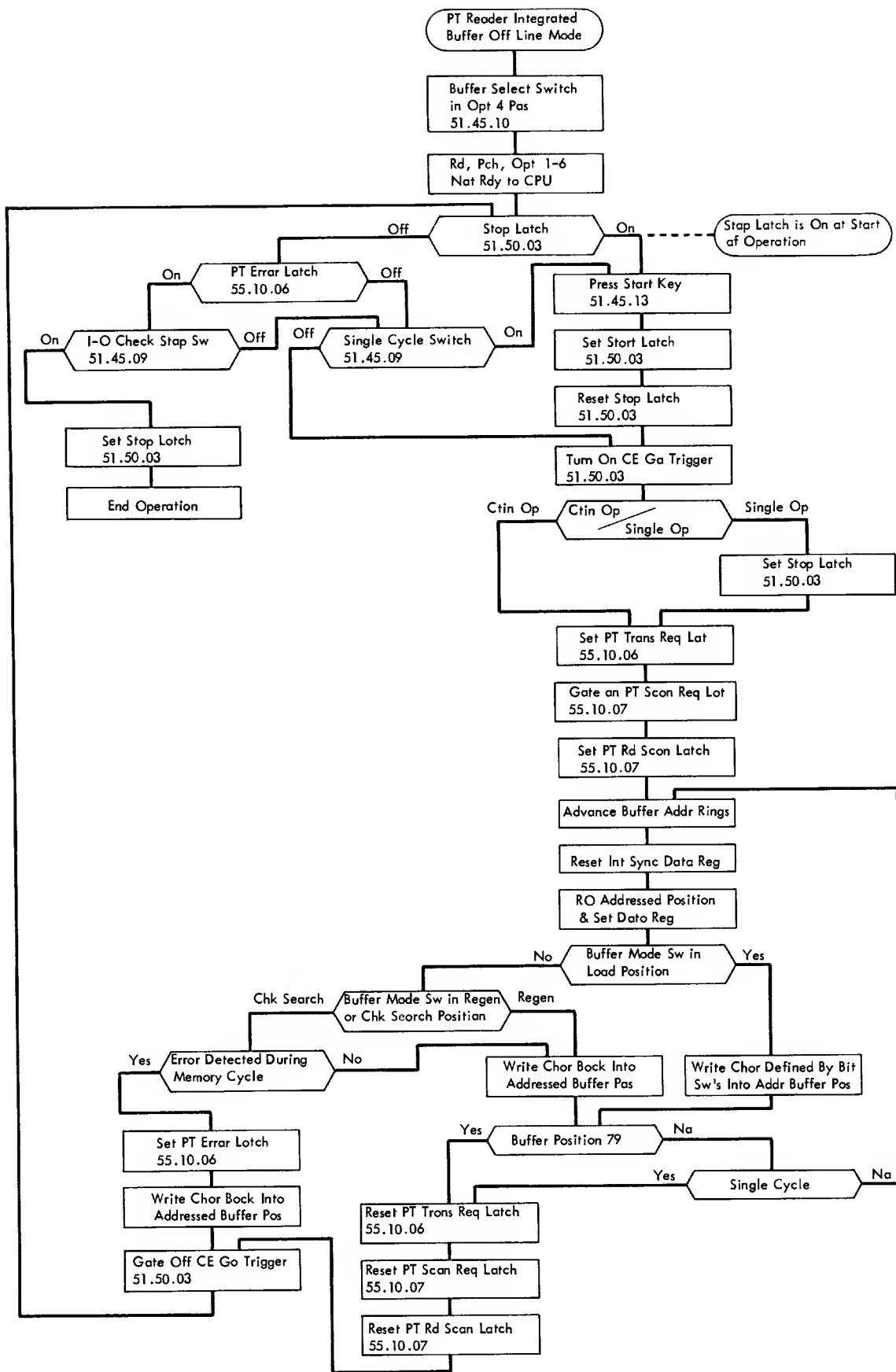


Figure 9.2-6 Paper-Tape Read Integrated Buffer Off-Line Mode

<u>Switch</u>	<u>Setting</u>	<u>Notes</u>
Single cycle	On	The operation stops after each memory cycle.
	Off	The operation is under the control of the Op-control switch.
Bit switches	Off or on	Cause corresponding bits to enter into the buffer (see buffer mode, load position).

The start key must be pressed to start any variation of this operation. Once the operation starts, it is stopped by the buffer-mode, single-cycle, and Op-control switches.

#### 9.2.06 Paper-Tape Read-Integrated-Buffer Off-Line Mode

The paper-tape integrated-buffer off-line mode allows paper-tape (PT) storage to be scanned for display, for error searching, or for entering new information from the CE panel bit switches. The entire integrated synchronizer and all associated units are removed from CPU control. The PT reader is not set in motion.

Operation (Figure 9.2-6)

<u>Switch</u>	<u>Setting</u>	<u>Notes</u>
Off line	On	-
Off-line mode	Integrated buffer	Removes the integrated buffer and associated units from CPU control.
Buffer select	Opt 4	-
Check stop	On	Stops the operation after the scan in which the error occurs.
	Off	Ignores errors (see buffer-mode switch).
Op control	Single op	Stops the operation after each scan (80 memory cycles).
	Ctin op	Causes repetitive scanning.
Carriage	-	No effect.
Buffer mode	Chk sch	The operation stops after the memory cycle in which the error occurs.
	Regen	Each position reads out and is regenerated.
	Load	Places the characters that the bit switches define into PT storage.





<u>Switch</u>	<u>Setting</u>	<u>Notes</u>
Single cycle	On	The operation stops after each memory cycle.
	Off	The operation is under control of the op-control switch.
Bit switches	Off or on	Insert bits when the buffer-mode switch is in the load position.

Pressing the start key starts the operation. After the operation starts, the buffer-mode, single-cycle, and Op-control switches control when it stops. Each scan is a full 80-memory-cycle scan instead of the normal serial scan that can vary in length.

#### 9.2.07 Print Integrated-Buffer Off-Line Mode

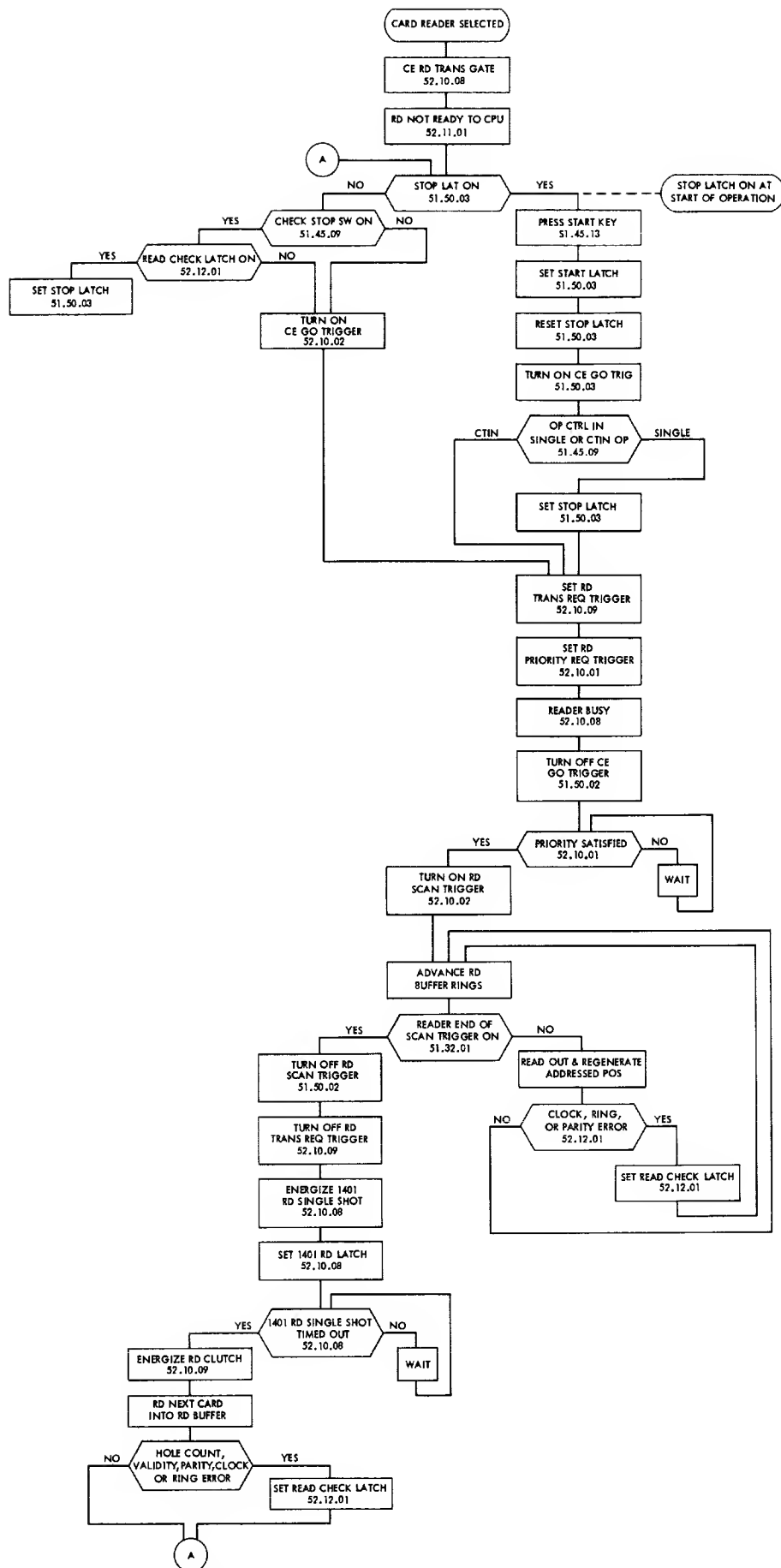
##### Function

The print integrated-buffer mode is similar in function and operation to the integrated-buffer mode. The print buffer circuitry is removed from CPU control. No information prints, because the printer is not set in motion. The print buffer is scanned under control of the other switches on the CE panel.

##### Operation (Figure 9.2-7)

The variations of this operation are set up as follows:

<u>Switch</u>	<u>Setting</u>	<u>Notes</u>
Off line	On	-
Off-line mode	Print buffer	-
Buffer select	-	No effect.
Check stop	On	Stops the operation after the scan in which error occurs.
	Off	Causes errors to be ignored.
Op control	Single op	Stops the operation after each scan.
	Ctin op	Causes continuous scanning.
Carriage	-	No effect.
Buffer mode	Chk sch	The operation stops after the cycle in which the error occurs.
	Regen	Each position reads out and is regenerated.



<u>Switch</u>	<u>Setting</u>	<u>Notes</u>
	Load	Allows the characters that the bit switches define to enter into the buffer.
Single cycle	On	The operation stops after each memory cycle.
	Off	The operation is under the control of the op-control switch.
Bit switches	Off or on	Insert bits when the buffer-mode switch is in the load position.

### 9.2.08 Card-Reader Selected Off-Line Mode

#### Function

In the selected-read buffer-off mode, the read buffer is removed from CPU control without affecting the other integrated buffer units. A transfer scan is taken. When the scan is complete, the card reader sets in motion to read the next card into the buffer. The operation is under control of the Op-control switch.

#### Operation (Figure 9.2-8)

This operation is set up as follows:

<u>Switch</u>	<u>Setting</u>	<u>Notes</u>
Off line	On	-
Off-line mode	Selected buffer	-
Buffer select	Reader	Removes only the reader from CPU control.
Check stop	On	Stops the operation after the card in which an error occurs.
	Off	Causes errors to be ignored.
Op control	Single op	Stops the operation after each card feeds.
	Ctin op	Causes continuous operation until reader becomes not-ready (see check stop).
Carriage	-	No effect.
Buffer mode	-	No effect. Data are regenerated during the transfer scan
Single cycle	-	No effect.
Bit switches	-	No effect.

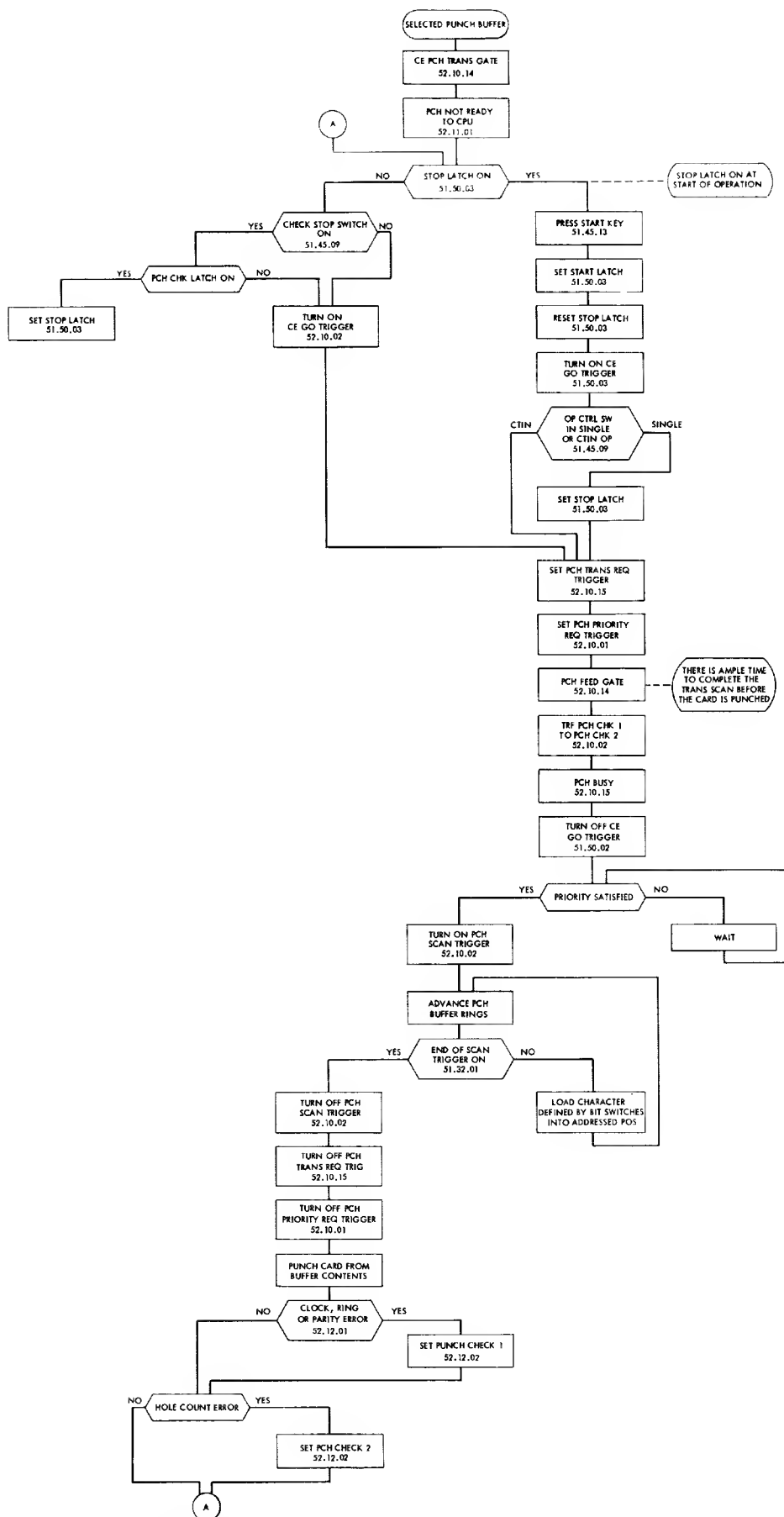


Figure 9.2-9 Punch Selected Off-Line Mode

### 9.2.09 Card-Punch-Selected Off-Line Mode

#### Function

This operation takes a punch transfer scan to fill punch storage. After the transfer scan, the record is punched into a card. Punch storage is removed from CPU control without affecting the other integrated buffer units.

#### Operation (Figure 9.2-9)

Because the other integrated-buffer units are still under CPU control, a complete scan must be taken each time an operation starts to free the buffer rings for other operations. This means that single-cycle and check-search operations cannot be performed in this mode. The selected punch-buffer off-line mode is set up as follows:

<u>Switch</u>	<u>Setting</u>	<u>Notes</u>
Off line	On	-
Off-line mode	Selected buffer	-
Buffer select	Punch	Removes only the punch from CPU control.
Check stop	On	Stops the operation after the card in which an error occurs.
	Off	Causes errors to be ignored.
Op control	Single op	Stops operation after each card.
	Ctin op	Causes continuous operation until the punch becomes not-ready.
Buffer mode	-	No effect. Characters are always loaded from bit switches during the transfer scan.
Carriage	-	No effect.
Single cycle	-	No effect.
Bit switches	Off or on	Insert bits during transfer scan.

Cards must be run into the punch to put the punch in a ready status. The start key is pressed to initiate the operation. Once the operation starts, the operation continues under control of the Op-control and check-stop switches.

### 9.2.10 Paper-Tape-Reader-Selected Off-Line Mode

The paper-tape-reader-selected off-line mode removes the PT buffer and the PT reader from CPU control for testing and troubleshooting. A PT transfer scan is taken, and when this scan is complete, the PT reader is signalled to begin reading. As each character is received from the PT reader, a serial scan is taken to store the incoming character.

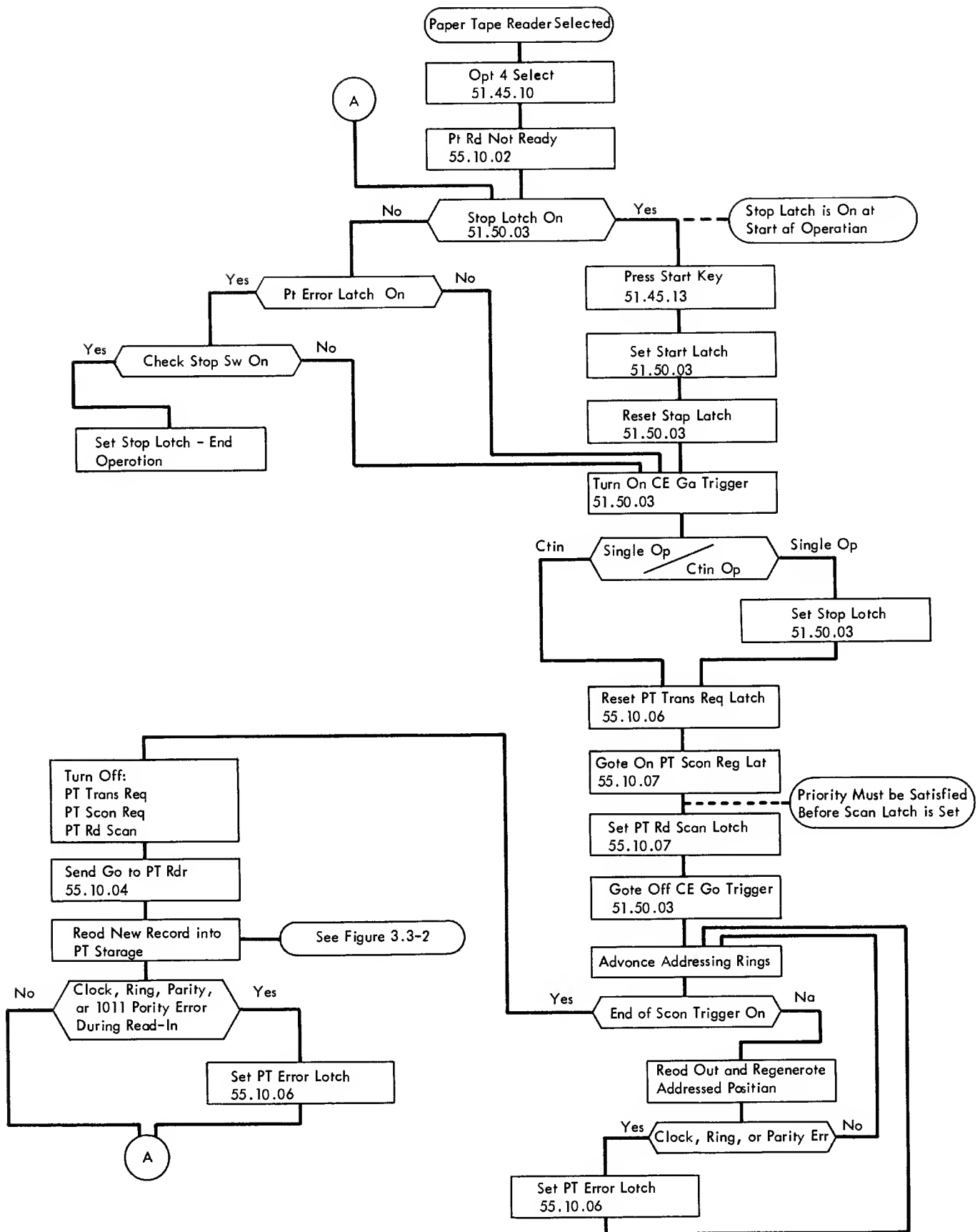


Figure 9.2-10 Paper Tape Reader Selected Off Line Mode

## Operation (Figure 9.2-10)

<u>Switch</u>	<u>Setting</u>	<u>Notes</u>
Off line	On	-
Off-line mode	Selected buffer	-
Buffer select	Opt 4	Opt 4 (PT reader) removes from CPU control.
Check stop	On	Stops the operation after the PT record in error.
	Off	Causes errors to be ignored.
	Single op	Stops the operation after each PT record.
Op control	Ctin op	Causes continuous operation until the PT reader becomes not-ready.
Carriage	-	No effect.
Buffer mode	-	No effect. Data are regenerated during transfer scan.
Single cycle	-	No effect.
Bit switches	-	No effect.

The start key starts this operation. Once the operation starts it continues under control of the Op control and check stop switches.

### 9.2.11 Printer-Selected Off-Line Mode

#### Function

This operation causes a print-transfer scan. During the transfer scan, the character defined by the bit switches is loaded into all print storage positions. The record is then printed. Either single lines or continuous printing can be started, depending on the setting of the Op-control and check-stop switches. Print storage is the only unit removed from CPU control.





## Operation (Figure 9.2-11)

The operation is set up as follows:

<u>Switch</u>	<u>Setting</u>	<u>Notes</u>
Off line	On	-
Off-line mode	Selected buffer	-
Buffer select	Printer	Removes only the printer from CPU control
Check stop	On	Stops the operation after the print line in which an error occurs.
	Off	Causes errors to be ignored.
	Single op	Stops the operation after each print line.
Op control	Ctin op	Causes continuous operation until the printer becomes not-ready.
	Sng space	The automatic-carriage single-spaces after each print line.
	Dbl space	The automatic carriage double spaces after each print line.
Carriage		
Buffer mode	-	No effect. Characters are always loaded from bit switches during the transfer scan.
Single cycle	-	No effect.
Bit switches	Off or on	Inserts bits during the transfer scan.

### 9.2.12 Oscilloscope Sync Generation

Two sets of switches on the CE panel define the oscilloscope sync pulses that are available at the sync hub. Either a print-buffer or integrated-buffer address can be selected, depending on the setting of the buffer-select switch. These pulses are always available whether operating on line or off line.

The customer engineer can further condition the sync pulses by adding a desired signal to the circuitry on logic 51.50.04. Any line labelled CE Input can have additional sync timing.

### 9.3.00 TAPE-ADAPTER UNIT CE PANEL (FIGURE 9.3-1)

The CE panel on TAU consists of: a pluggable control panel, fifteen switches, and an indicator panel. The CE panel duplicates all of the request signals to TAU in addition to making manual data lines available. All operations in TAU occur in a normal manner.

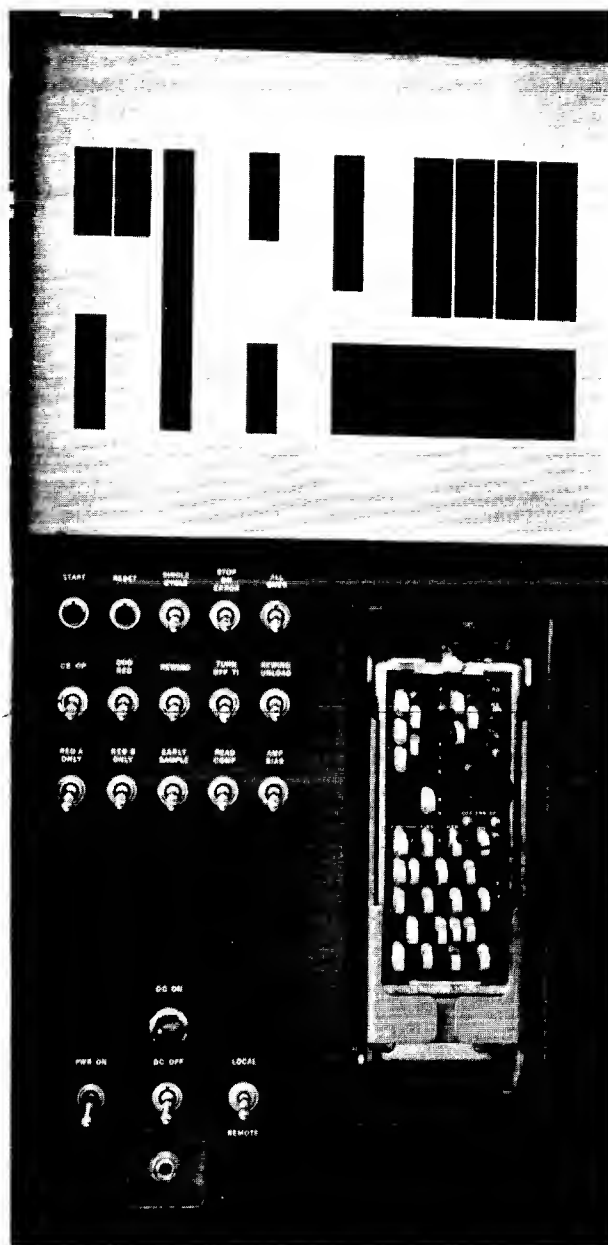


Figure 9.3-1 Tape-Adapter Unit CE Panel

## Switches

**Stop on Error.** When this switch is active, a TAU manual stop-on-error-line is conditioned. This line checks the condition of the TAU error latch and the R/W vertical redundancy check (VRC) latch. If the error latch is ON, the stop-on-error line blocks the inputs, and resets to the read registers A and B. If the R/W VRC latch is ON, the stop-on-error line blocks the inputs, and resets to the R/W register. At the end of the operation, these registers contain the character or condition that caused the error. The stop-on-error line also prevents further TAU operations from being performed by keeping the busy line active. The stop-on-error line is conditioned for the following errors, if the stop-on-error switch is active:

1. Read longitudinal redundancy check register (LRCR) error
2. Skew error
3. No-echo error
4. R/W register VRC error
5. Write A-register VRC error
6. Compare error.

Because TAU checks for a compare error at the same time that it resets the read registers, the manual stop-on-error line gates an earlier read-clock pulse, and the stop-on-error line blocks the reset to the read registers.

**CE Operation.** If this switch is active, TAU is under control of the CE panel. The CE Op switch must be ON to condition all of the other switches in CE operation except stop-on-error. The CE operation switch lights the tape off-line indicator on the 1415 console.

**CE Reset.** If this switch is active in CE operation, the CE control circuits reset, and the TAU reset line is conditioned. The TAU reset line resets the circuits in TAU to their normal status.

**CE Start.** Pressing the start button in CE operation, turns on the start latch. The start latch output generates a CE pulse. This CE pulse and the pluggable-control panel develop the manual-request signals for TAU. Releasing the start button resets the start latch. The busy line at the end of the TAU operation generates the next CE pulse.

**CE Single Cycle.** The CE single cycle switch blocks the generation of a CE pulse by the busy line. The start button must be pressed for each manual request signal to TAU.

**Write All Ones.** When this switch is active in CE operation, the write-data lines are conditioned to write all seven bits of a character each write cycle.

CE Rewind. If this switch is active in CE operation, a rewind call signal is sent to the TAU. The TAU operates in a normal manner when it receives this request signal, and rewinds the selected tape unit.

CE Rewind Unload. If the CE rewind-unload switch is active in CE operation, a rewind-unload call signal is sent to the TAU. The TAU operates in a normal manner when it receives this request signal, and rewinds and unloads the selected tape unit.

Turn Off TI. When this switch is active in CE operation, a manual-request signal is generated to turn off the tape-indicator in the selected tape unit.

Odd Redundancy. If the odd redundancy switch is on in CE operation, a manual odd-redundancy-call signal is generated to turn ON the odd-redundancy latch in TAU. This odd-redundancy latch conditions all TAU VRC's for the odd-redundancy operation. If this switch is not active, the TAU VRC's are conditioned for the even-redundancy operation.

Register A-Only. When this switch is active in CE operation, conditions are set up to force TAU to use the A-read register for data transmission. The switch conditions the register-A-only line, that blocks the A-read-register vertical-redundancy check. Because the A-read register cannot indicate whether it is invalid, the data for the R/W register and the LRCR always comes from the A-read register.

Register B-Only. If this switch is active in CE operation, conditions are set up to force TAU to use the B-read register for data transmission. This switch conditions the register-B-only line that causes an A-read-register vertical-redundancy error every read cycle. Because the A-read register indicates an error, the data for the R/W register and the LRCR always comes from the B-read register.

Compare Check. The compare-check switch in CE operation compares the A-read register to the B-read register during a read operation. Because the registers are compared as a function of the normal write operation, this switch is not needed during writing.

Early Sample. If the early-sample switch is active in CE operation, the skew-gate latch in TAU turns ON one read count earlier during read and write operations. The skew-gate latch turns ON early to enable a more critical check of the skew within each character.

Amplifier Bias. When this switch is active in CE operation, the acceptance levels of the TAU final amplifiers are altered. In a read operation, the final amplifiers are conditioned to accept data at the normal write-operation levels. In a write operation, the final amplifiers are conditioned to accept data at the normal-read operation levels.

Pluggable Control Panel. The pluggable control panel enables customer engineers to program the TAU operations manually. By using the control panel, a customer engineer can select tape units, generate request signals, vary the length

of the tape records, and vary the bit configurations for each of five characters. The CE control circuits include three rings or counters to condition the hubs of the control panel. These three rings (instruction, special, and character counter) operate independently.

**Instruction Ring.** The instruction ring is a four-stage counter that the CE reset line resets and steps when the busy line is deconditioned. The reset line turns the first stage ON, and turns the other three stages OFF. As a result of the reset, the output from the first stage is available immediately after reset time. Instruction ring 1, instruction-out, and select-out hubs of the control panel are conditioned at this time. These hubs select a tape unit and request a TAU operation. When the CE start button is pressed, the CE pulse generates the request signal to TAU. The busy line is conditioned when TAU accepts the request signal, and it remains conditioned throughout the operation. When the operation ends, the busy line is deconditioned. At this time, the instruction ring is stepped and another CE pulse is generated to start the next operation. The instruction ring operates in a loop (1, 2, 3, 4, 1, 2, 3, 4, 1, etc.) until the CE reset switch is activated, or until the special ring is conditioned, or until the single-cycle switch is activated.

**Special Ring.** The special ring can be used only following an error in TAU. The control-panel error and special-routine hubs must be connected to condition the special ring. This special two-stage ring is used for special CE programming when an error is detected. Example: backspace and read, backspace and write, backspace and erase, etc.

**Character Counter.** The thirteen-stage character counter controls the variable-length record and the sequence of the five characters. The outputs of the character counter condition the character hubs, and count the hubs of the control panel.

**Select Out.** When these hubs are conditioned by the instruction ring and special ring, they select a tape unit for operation.

**Select In.** These ten positions of the control panel correspond to the ten tape units that TAU operates. The hubs are wired from the Sel Out hubs and must be conditioned for TAU operation.

**Instruction Out.** When the instruction-out hubs are conditioned by the instruction ring and the special ring, they generate request signals to TAU.

**Instruction In.** These five hubs are wired from the Inst Out hubs and are used with the CE pulse to develop the request calls to TAU.

**Off, Error, Special Routine.** These three hubs control the operation of the special ring. If the special routine hub is wired to the error hub, an error condition in TAU starts the special ring. If the off hub is wired to the special routine hub, the special ring operation is blocked.

**Count Hubs.** These hubs control the timing of the disconnect call signal to TAU. The count (CNT) hub must be wired from one of the count number hubs (1, 5,

80, 320, 1000, or continuous). The count number hubs are conditioned by the character counter.

Character Hubs. Five sets of character hubs are available on the control panel. Any bit configuration desired can be wired for each of the five characters. The sets of character hubs are controlled by the character counter. If the number of characters that is called for exceeds five, the output from the character hubs is repeated. Example: 1, 2, 3, 4, 5, 1, 2, 3, 4, 5, 1, 2, etc.

Indicator Panel. The indicator panel contains a visual indication of the status of the control circuits in TAU and of the CE control unit. There is an indication for each register, counter, clock and ring, as well as most of the control latches.

## 10.0.00 POWER SUPPLIES

The IBM 1410 Data Processing System requires a 3-phase 208 or 230 volt ac source of voltage. The main-line voltage feeds to the 1411A frame of the CPU. The power-on key on the console controls the tape frames, RAMAC<sup>®</sup> files, reader-punch, and printer except when they are off-line for independent operation. The independent control must be manually set for the peripheral equipment.

Power is brought to a circuit-breaker-distribution panel (Figure 10.0-1) in the 1411A frame of the CPU. The ac voltage feeds from the circuit breakers into ferroresonant regulators where it is changed to a regulated ac voltage. These regulated ac voltages are distributed to the dc supply modules, where ac is converted into regulated dc voltages.

The circuits that use the incoming voltage without regulation, motors, etc. feed directly from the source through the circuit breakers and to the units.

The various power supply voltages that are distributed from the main power source are: +6V dc, -6V dc, +12V dc, -12V dc, +30V dc, -36V dc, -48V dc, +60V M, 115V ac and 208V ac.

These voltages from the individual power supplies are brought to a laminar bus in each frame. The laminar bus makes all the various voltages available for easy access to the SMS card bases for wiring.

### 10.1.00 REGULATION

The ferroresonant regulators that the 1410 power supply uses consist of special transformers and capacitors. The capacitors and the transformers are packaged into separate modules for flexibility in mounting. The regulators are available in several output ratings, ranging from 250 to 1840 watts.

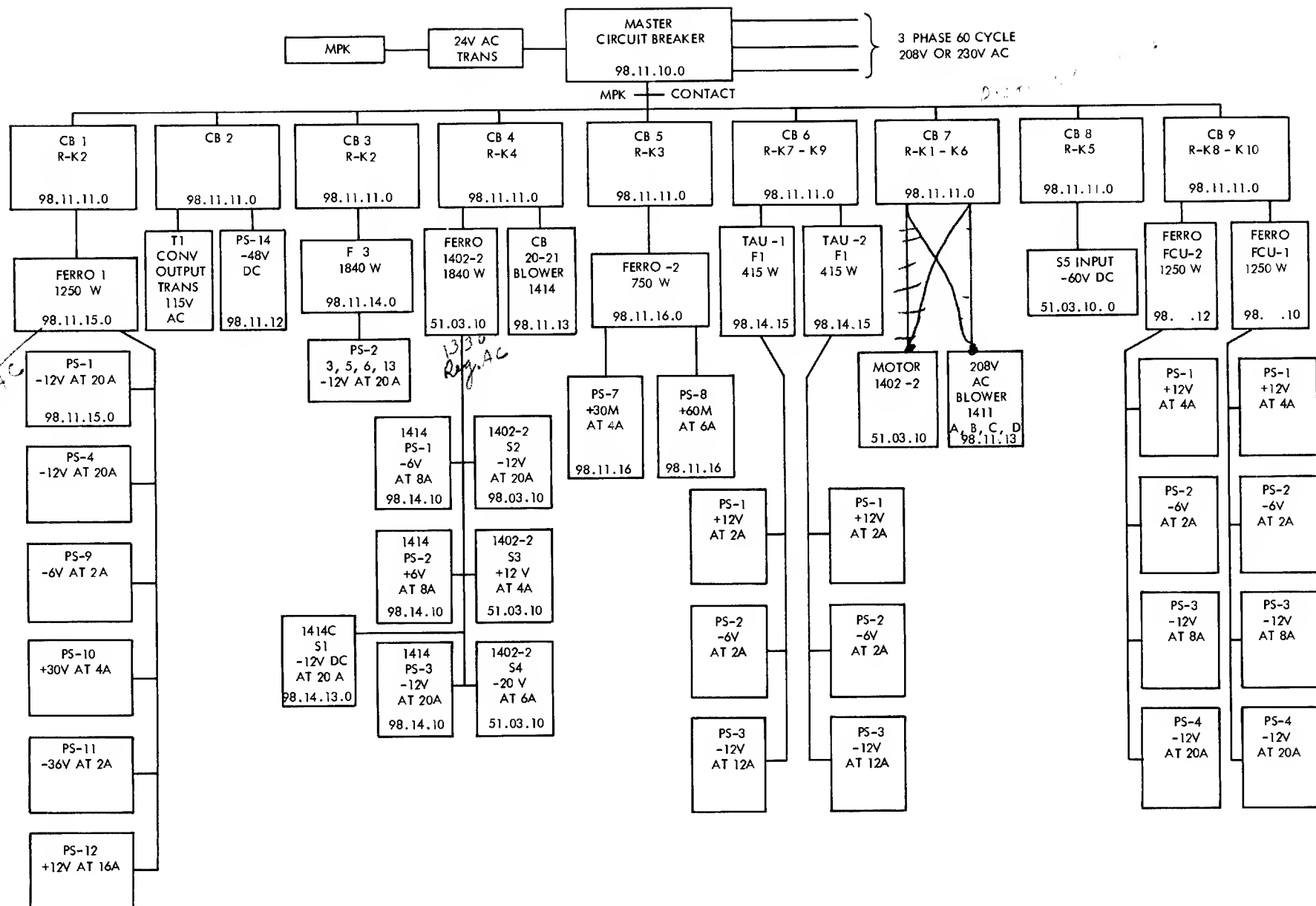
The ac regulator changes the line voltage to a regulated voltage of approximately 133V ac for the dc power-supply modules.

The dc power-supply module consists of an isolation transformer, solid-state rectifiers, associated filter networks, and a magnetic-type circuit breaker for overcurrent protection. The dc power supply incorporates a series regulator (that consists of an SMS pluggable-amplifier card), power transistors, and filters. These provide a dc voltage with  $\pm 2\%$  regulation.

### 10.2.00 SPECIAL-VOLTAGE POWER-SUPPLY UNIT

Special voltages are required to supply core-driver circuits. Two voltages are developed from the two supplies 7 and 8 that feed through the ferroresonant regulator 2 from circuit breaker 5. Both power supplies 7 and 8 have a separate control for setting their output voltage. PS7 supplies +25V dc at 4 amperes and PS8 supplies +60V dc at 6 amperes.

Figure 10.0-1 IBM 1410 Power Distribution





445



### 10.3.00 POWER-ON SEQUENCE (FIGURE 10.3-1)

The master-circuit breaker that is located on the 1411A power-supply frame controls the incoming power to the system. When the master-circuit breaker is ON, power is supplied to a 24V ac transformer. This transformer (T2) supplies the power to pick MPK relay, that closes the MPK contacts 1, 2, 3.

With power on at the circuit-breaker panel, pressing the power-on key on the console starts the power-on sequence. (Figure 10.3-2)

### 10.4.00 EMERGENCY OFF

Pulling the emergency-off switch, drops relay MPK, and opens the circuit at the multiple-circuit breaker. MPK points 1, 2, 3, are in series with the CB panel. This also drops the power to other peripheral equipment that have their own line cords.

The emergency-off switch is mechanically interlocked so that once it has been operated, it can not be reset from outside the console.

To reset the emergency-off switch, the control-unit cover must be removed from outside the console, and the contact locking spring must be lifted.

Relay 76 provides the emergency power control for special features that can be added to the system.

### 10.5.00 DC-OFF (FIGURE 10.5-1)

Pressing the dc-off key on the console drops all the dc power to the units. The system-power-off, memory-heater, and file-ready relays are not affected by the dc-off operation. The dc-off sequence starts when the machine drops R5, and then steps down to a complete dc power-off condition on the system. (Figure 10.5-2)

Other conditions that initiate dc-off are:

1. Loss of an output of a ferroresonant regulator. This causes the corresponding sense relay to drop. This condition opens the hold circuit to R5 dc-off relay.
2. The tripping of a series regular CB. If a series regulator CB in any frame trips, R53 is picked up, and two indicator lights come on. One light indicates that a power supply CB tripped, and the other shows the location of the power supply.
3. If the memory-array temperature goes above or below the specified limits, the thermal switch closes. This condition picks R69 and also drops R5.

Pressing the power-on key restores the dc power and the sequence for power-on takes place in the same order as a normal power-on operation.

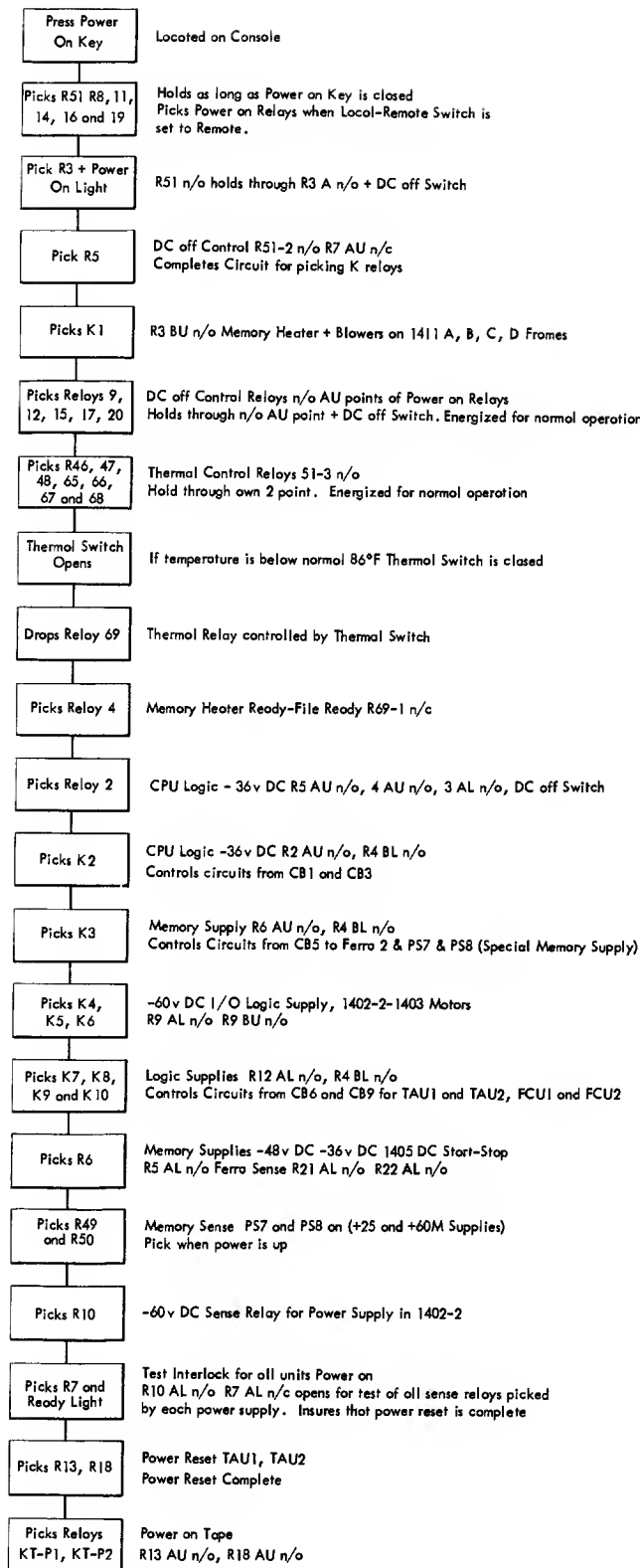


Figure 10.3-2 Power On

**Figure 10.5-1 DC Off Timing Chart**

[illegible]

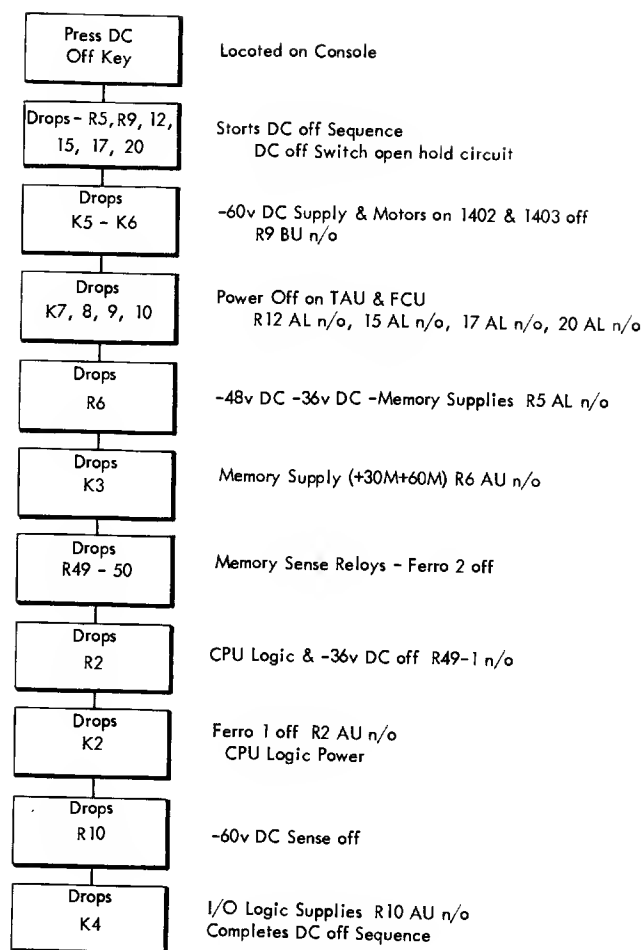


Figure 10.5-2 DC Off

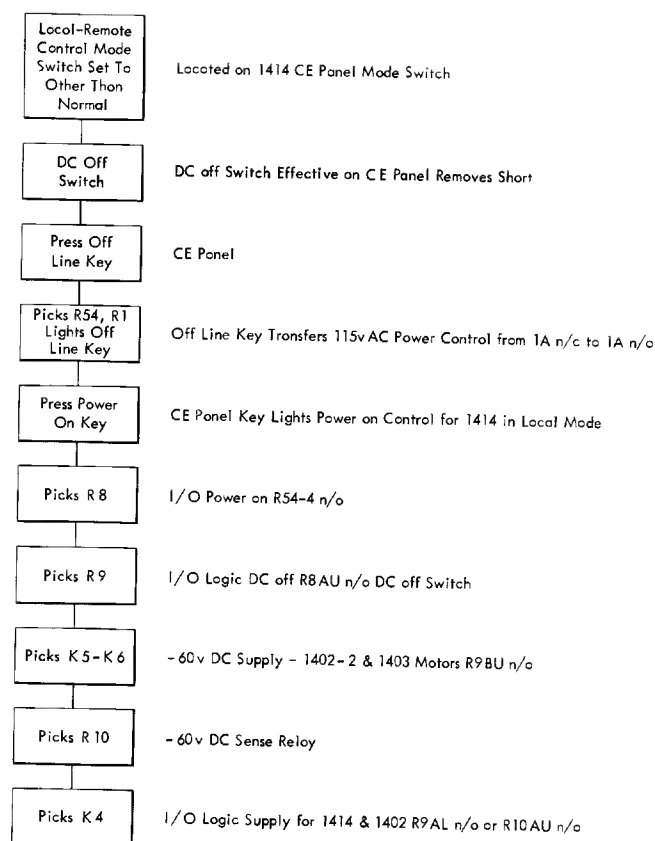


Figure 10.7-1 Input-Output Synchronizer (Power On)

## 10.6.00 POWER-OFF

Pressing the power-off key removes all ac and dc power from the system. The power-off sequence starts by dropping the R3 system power-off relay. The R3 AL drops the R5 dc-off relay. This condition starts the machine into a dc-off condition.

Power-off is initiated by the following conditions:

1. Pressing the power-off key on the console.
2. The blower CB trips.
3. A thermal switch in one of the frames opens.

When either the blower CB trips, or a thermal switch opens:

1. One of these thermal relays drops: R46, 47, 48, 65, 66, 67, 68.
2. A latch relay, R45, picks and latches up.
3. Two lights light:
  - a. A thermal light indicates that the trouble is thermal.
  - b. A location light indicates the frame in which the trouble is located.
4. 45-1 drops R3 (power-off relay).
5. 45-2 opens and prevents resetting of the thermal relays.
6. 45-3 lights the thermal lights.
7. 45-4 prevents any of the lights from coming on with an initial power-on.

## 10.7.00 LOCAL REMOTE OPERATION - I/O (FIGURE 10.7-1)

When the off-line mode switch on the 1414 I/O synchronizer is set in any other position than normal, the power supply and on-line controls from the 1411 CPU are not effective. This switches the I/O units to an independent operation. Pressing the off-line key on the 1414 console picks relays 54 and 1 in the 1411 CPU. This transfers the 115V ac supply from R4 BL, through 1AN/C, directly to the output of the transformer, and through 1AN/O. R54-1 N/C opens the circuit to short out the power-off switch on the 1414 and to make it operative. This relay switches the dc control from the console unit to the 1414 CE panel on and off switches. The light on the console indicates the status of the units for local or remote operation. The off-line light is ON when the unit is selected for local operation. In this mode the power for the synchronizer is under the control of the power-on and off keys that are located on the CE panel of the synchronizer.

If the switch is transferred from remote to local after the power is on the system, the dc power is not lost to the synchronizer. Pressing the power-off key that is located on the CE console drops relays 9 (I/O logic dc-off) and 24 (ferrosense).

#### 10.8.00 TAU OFF-LINE OPERATION-LOCAL-REMOTE SWITCH (FIGURE 10.8-1)

The local-remote switch on the CE panel of the TAU units selects each unit for on-line or off-line operation. With the switch set to remote mode, the unit is on line with the system. In local mode, the unit is off line and has independent dc-on, dc-off operation from the CE panel.

#### 10.9.00 FILE CONTROL UNIT (FCU) OFF-LINE OPERATION-LOCAL-REMOTE (FIGURE 10.9-1)

The local-remote switch on the CE panel provides manual control (similar to the TAU) for selecting the FCU for on-line-with-the-system operations, or for off-line-for-checking-and-servicing operations.

#### 10.10.00 POWER-FAULT INDICATORS

The loss of power from any ac or dc supply is detected by the picking of relay 53 in the 1411A frame. Picking of relay 53 opens the circuits to dc-off control for the system, and indicates the type of failure and unit location by lights on the 1411A control panel. The failure can be classified as one of the following types.

##### 10.10.01 Thermal Reset

This condition latches up thermal relay 45. This isolates the power-on circuit until the thermal-relay latch is reset. Pressing the thermal-reset key on the 1411 CE panel-latch trips relay 45, and restores the circuitry to normal.

##### 10.10.02 Overcurrent Failure

This type of failure trips the circuit breaker on the particular power-supply module that is involved. This condition is caused by overloading the power supply because of a circuit-component failure. A light on the control panel of the 1411A unit indicates the general location of the power unit.

##### 10.10.03 Overvoltage Failure

The overvoltage type of failure is detected by an overvoltage protection device that is set for a maximum voltage range. When the voltage-protection device senses an overvoltage condition, a circuit breaker trips and power turns off. An indicator shows the section where the failure occurred.

#### 10.11.00 MARGINAL CHECKING

Marginal checking in the 1410 system can be accomplished on any frame. The marginal-check circuitry is brought to the CE panels of the units and terminates at a jack. Through jack connections, the marginal-check portable-power supply can be connected to a particular frame for checking. Controls on the portable supply perform the marginal checking.

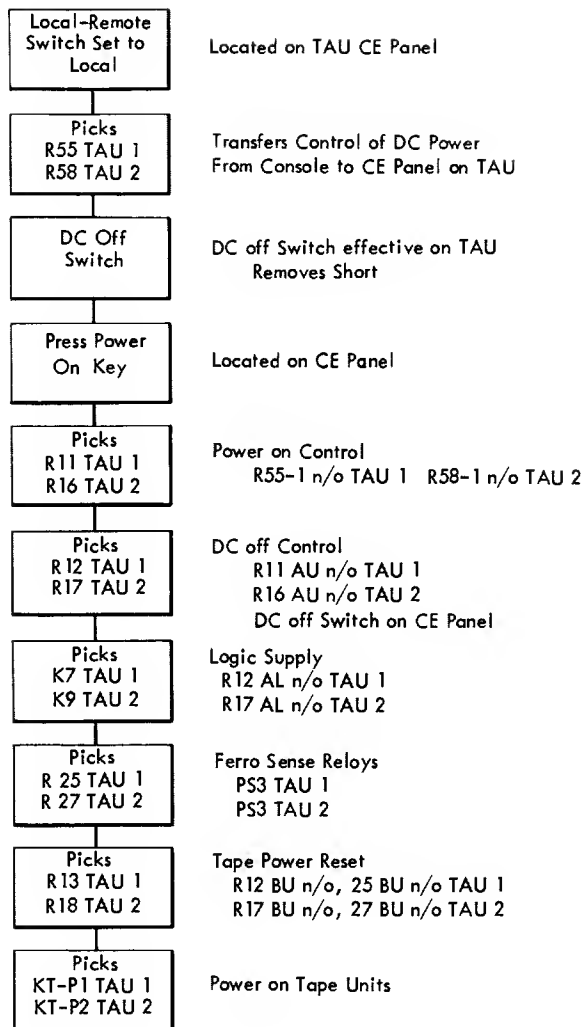


Figure 10.8-1 Power On, Tape Units

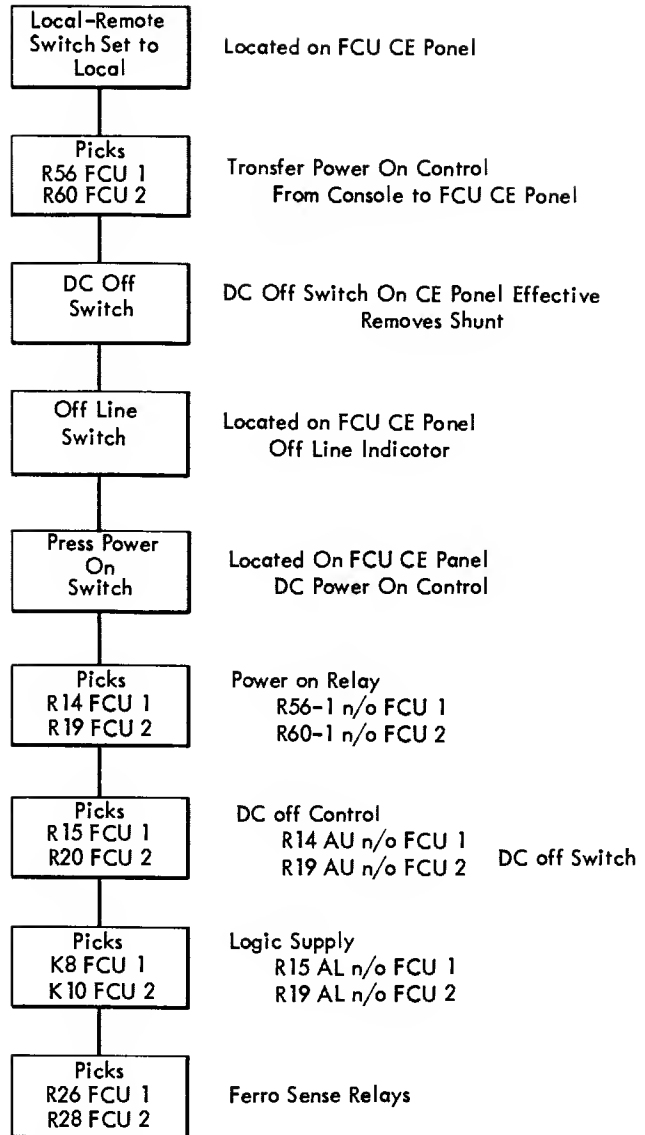


Figure 10.9-1 Power On, File Units



## 10.12.00 PORTABLE MARGINAL-CHECK POWER-SUPPLY UNIT

The application of the marginal-check unit to a system permits the customer engineer to change a particular voltage by  $\pm 3V$ . For example, +12V can be varied from a +9V to a +15V to assist in locating marginal or intermittent failures.

The portable marginal-check unit is designed for use with machines that have the marginal-check circuitry wired to jack receptacles on the CE panels.

The unit operates on 115V ac and has a series regulator built in to supply a normal regulation of  $\pm 2\%$  in reference to ground.

The remote control is attached to the power supply unit by a long cable, so that the control can be carried around the machine while marginal-checking is being performed.

Remote control permits full control of the marginal power-supply unit. The variable control has an off position for turning the output OFF, and a potentiometer to control the output voltage from 0V to 3V. A toggle switch on the unit gives a setting to select either buck or boost voltage.

The power unit has a power-supply attachment cord for input voltage from any available 115V ac outlet. The output cord has a plug to connect the power supply to the jack receptacle on the CE panel of the unit by means of the marginal-check jack receptacles.

The unit is protected from overload voltage by a 6-ampere circuit-breaker that is located on the component panel in the power-supply unit. The marginal-check unit circuit-breaker trips, if the remote control switch is set to the boost position, and if the jack plug is not inserted in a marginal-check jack receptacle when power is ON.

The main-line switch is located on the top near an indicator light. The light gives a visual indication that the power is ON when the unit is in use.

A safety control is built into the marginal check unit. The control insures that the potentiometer is set at zero when the unit is attached. This prevents any surge of current into the transistor circuitry that could result in damage to the components of that circuit.

When the power supply is first set up and connected to the machine marginal-check jack, it does not have any output until the remote control has been set to zero. This step picks the RY1 relay that opens the power-supply output shunt. The RY1 relay is self-holding until the jackplug is disconnected, or until the main-power switch on the unit is turned off. Caution: Always extract the jackplug from a marginal-check jack receptacle with the control potentiometer set at zero.

Figure 10.12-1 is a block diagram of marginal-check power supply. Figure 10.12-2 illustrates the theory of operation when the marginal-check unit is connected in series with the SMS power supply. The rectified dc voltage acts as a battery that is in series with the SMS power supply. The series regulator effects a controlled impedance in series with the rectified dc voltage. When in boost-

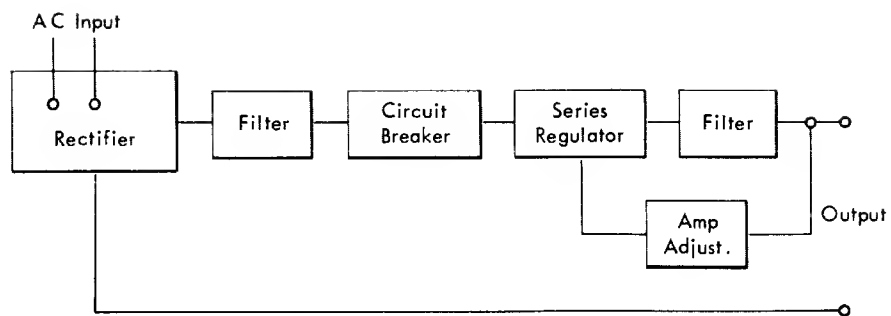


Figure 10.12-1 Marginal-Check Power Supply (Portable Unit)

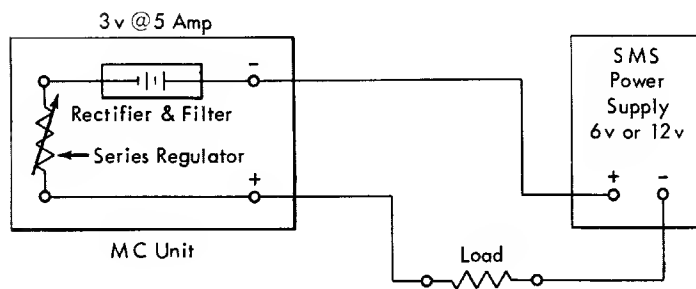


Figure 10.12-2 Marginal-Check Unit In Series with SMS Power Supply

output mode, the impedance is low. This allows the dc voltage to add to the load voltage. When in buck-output mode, the impedance is high. This condition causes all the rectified dc voltage, and a maximum of 3 volts of the load voltage, to drop across the impedance. This results in a voltage reduction across the load. The amount of change effected by this unit depends upon the drive that the series regulator induces within the maximum-load limit of 5 amperes.

The SMS series regulator-control card plugs into the receptacle in the power-supply unit.

Caution: When the unit is in use do not block the ventilating fan airflow through the ends of the power supply unit.

#### 10.13.00 POWER-UNIT LOCATIONS

Figure 10.13-1 shows the location on the front of the 1411A frame of the following: the power-supply unit, circuit breakers, cable connectors, K-relays, controls, indicator lights, and marginal-check jacks.

Figure 10.13-2 shows the location on the rear of the 1411A frame of the following: the relay gate assembly, terminal blocks, control transformers, and power supply.

#### 10.13.01 Frame and Chassis Designations

Figure 10.13-3 is a chart that shows the voltage designations for the laminated bus pins.

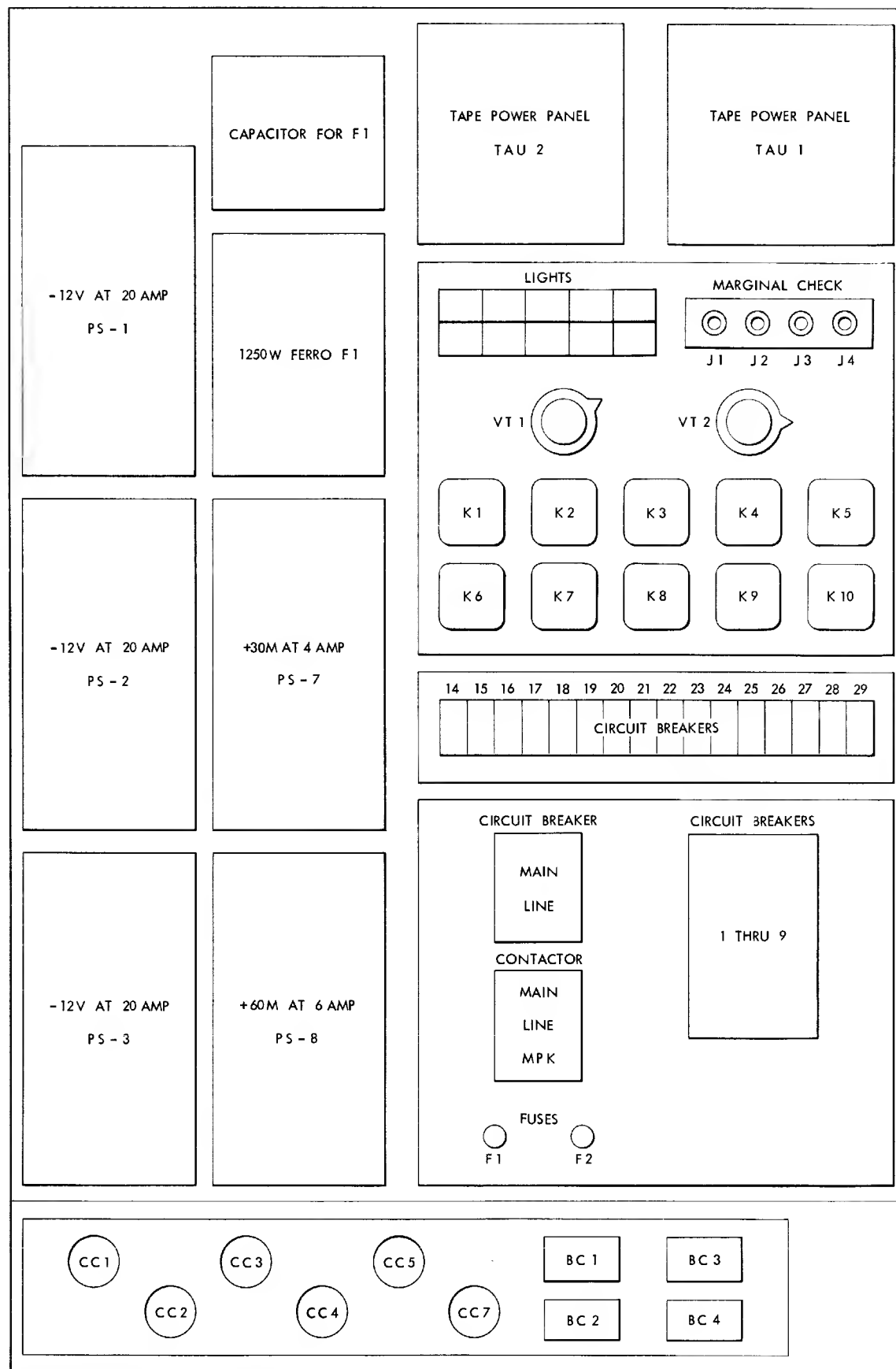


Figure 10.13-1 IBM 1411 A Frame Front

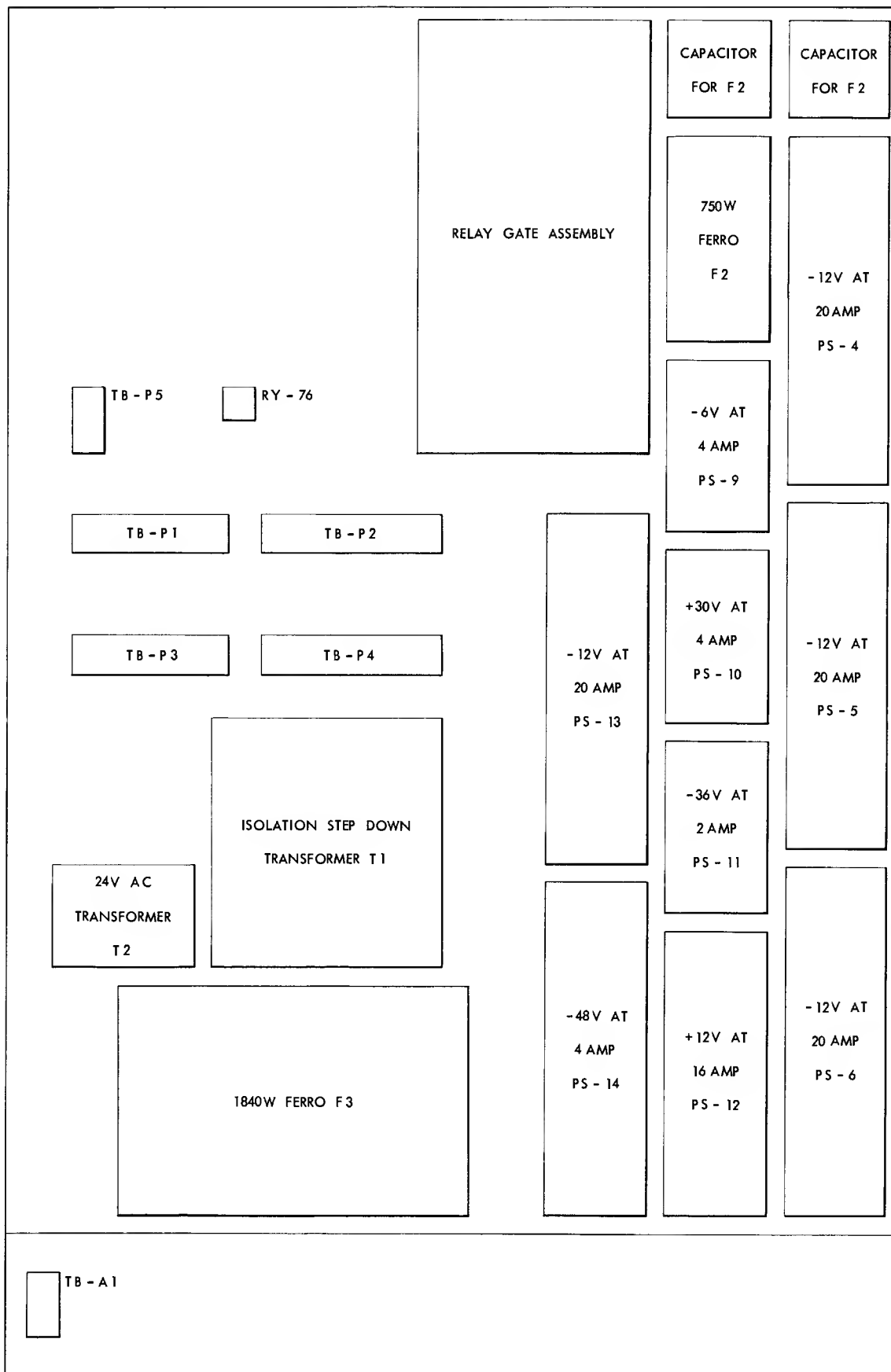


Figure 10.13-2 IBM 1411A Frame Rear

FRAME AND CHASSIS DESIGNATION																
LAMINAR BUS	11B2	11B3	11B4	11C1	11C2	11C3 11C4	11D1	11D2	11D3 11D4	14A1	14A2 14A3 14A4	14C *	14C **	14D		LAMINAR BUS
1-G	-12V			-12V			-12V				-12V	-12V		-12V		1-G
2-H	GND	GND	GND	GND			GND				GND	GND		GND		2-H
3-J		-12V	-12V			-12V		-12V		-12V			-12V			3-J
4-K		GND	GND			GND		GND		GND			GND			4-K
5-L	+12M			+12M	+12M	+12M	+12M	+12M	+12M	+12M	+12M	+12M	+12M	+12M		5-L
6-M	GND	GND	GND	GND	GND	GND	GND	GND	GND	GND	GND	GND	GND	GND		6-M
7-N		+12M	+12M		-12V				-12V	+6V	+6V	+6V	+6V	+12V		7-N
8-P	-6V	-6V	-6V		GND				GND	-6V	-6V	-6V	-6V	-6V		8-P
9-Q										GND	GND	GND	GND	GND		9-Q
10-R										+12V	+12V	-20V	-20V			10-R

\* = 1414C CHASSIS 4, 5, 6 AND ROWS E, F OF CHASSIS 3

\*\* = 1414C CHASSIS 1 AND ROWS A, B, C, D OF CHASSIS 3

Figure 10.13-3 Frame and Chassis Designation